Цифрлық әдіскер
ақпараттық-білім беру ортасы

ПРАКТИКАЛЫҚ САБАҚТАР

Python бағдарламалау тілі

Болашақ информатика мұғалімдеріне арналған Python негіздері: синтаксис, деректер құрылымдары, функциялар және алгоритмдік есептер.

- Синтаксис және деректер түрлері
- Шартты операторлар мен циклдер
- Тізімдер, сөздіктер, жиындар
- Функциялар және модульдер
- Файлдармен жұмыс
- Алгоритмдік есептер

ӘДІСТЕМЕЛІК НҰСҚАУ

Практикалық жұмыстар

Практикалық жұмыс 1: PyGame кітапханасы. PyGame кітапханасын орнату

Python тілі сізге графикалық интерфейспен қосымшаларды жасауға мүмкіндік береді, осы мақсат үшін түрлі графикалық кітапханалар қолданылады. Стандартты (Python стандартты жинағына кіреді) графикалық кітапхананың tkinter бөлімін қарастырайық. Ең алдымен PyGame, tkinter-мен жұмыс жасағанда, графикалық элементтердің қалғанын (виджеттер) орналастыратын негізгі (түбірлік) терезені жасау керек. Мәтінді енгізу, мәтінді шығару, ашылмалы мәзір және т.б. үшін әр түрлі виджеттер бар. Кейбір виджеттер (фреймдер) өз ішіндегі басқа виджеттерді топтастыру үшін пайдаланылады. PyGame кітапханасы. PyGame кітапханасы Python бағдарламалау тілінде ойын жасауға мүмкіндік беретін кітапхана. Базалық курсты аяқтағаннан кейін тілдің синтаксисін түсіну әлдеқайда жеңілірек болады. Python-да кеңейтілген және оны кез-келген функциямен толықтыратын әртүрлі кітапханалар бар. Бұл тілде интерфейсі бар ойындар жасауға арналған арнайы жасалған кітапхана PyGame кітапханасы. Осы курста біз қарапайым 2D ойындарын Python тілінде құрамыз. Біз ойынды анимациямен, спрайтпен (суреттермен), функционалдылықпен және графикалық интерфейспен жасаймыз. Кітапхананың өзі 2000 жыл. Шығарылды. Жазылған жоба Android-те де, компьютерлерде де қауіпсіз жұмыс жасай алады. Осылайша, сіз бірнеше құрылғыларда жұмыс істейтін ойындар жасап, сонымен бірге код жазуды дағдыланамыз. Осы кітапхана негізінде көптеген ойындар мен қосымшалар жасалды.

Python программасын орнату. Python.exe орнату файлын іске қосыңыз

1. PyGame кітапханасын орнату
2. Пуск→ Командная строка

- pip install pygame→Enter
- python→Enter
- imoprt pygame→Enter

Тапсырмалар:

- PyGame кітапханасын компьютерге орнатып көріңіз.
  Терминал (немесе командалық жол) арқылы pip install pygame командасын орындаңыз. Орнату процесінен кейін скриншот жасаңыз немесе нәтижені жазыңыз.
- PyGame-нің орнатылғанын тексеретін қарапайым Python бағдарламасын жазыңыз. import pygame print("PyGame сәтті импортталды!")
- PyGame кітапханасын пайдаланып, бос терезе ашатын бағдарлама жазыңыз. 640x480 пиксельді терезе ашылып, «Pygame терезесі» деген атау көрінуі тиіс. Үлгі: import pygame pygame.init() screen = pygame.display.set_mode((640, 480)) pygame.display.set_caption("Pygame терезесі") pygame.quit()
- Білім алушы өз құрылғысына PyGame орната алмай жатса, қандай себептер болуы мүмкін? Және оны қалай шешуге болады? Мәселе → Себеп → Шешу жолы түрінде сипаттау.
- PyGame орнатылса да, ModuleNotFoundError қатесі шықса не істеу керек? Python нұсқасы мен виртуалды орта мәселесін зерттеу.

Дескриптор:

- PyGame кітапханасын орната алады;
- PyGame кітапханасының орнатылғанын тексереді;
- PyGame кітапханасында жасалған программаны сақтау жолдарын көрсетеді. Сұрақтар:
- PyGame дегеніміз не?
- PyGame кітапханасы қандай мақсатта қолданылады?
- PyGame кітапханасының артықшылықтары қандай?

- PyGame кітапханасы қандай тілде жазылған?
- PyGame кітапханасын қолдану үшін қандай бастапқы білім қажет?
- PyGame қандай ойын түрлерін жасауға мүмкіндік береді?
- PyGame мен басқа ойын жасау кітапханаларының айырмашылығы қандай?
- PyGame кітапханасын орнату үшін қандай команданы қолданамыз?
- PyGame орнату үшін Python қандай нұсқасы қажет?
- PyGame-ды орнату кезінде қандай қателер болуы мүмкін?
- PyGame орнатылғанын қалай тексеруге болады?
- pip дегеніміз не?
- pip арқылы басқа қандай кітапханаларды орнатуға болады?
- PyGame орнатылғаннан кейін оны қалай импорттаймыз?
- "import pygame" командасы не үшін қажет?
- PyGame орнатуды Windows жүйесінде қалай жүргізуге болады?
- Linux жүйесінде PyGame қалай орнатылады?
- PyGame орнатылғанын командалық жолда қалай тексеруге болады?
- PyGame терезесін қалай ашамыз?
- PyGame терезесіне атау беруді қалай орындаймыз?
- PyGame кітапханасының артықшылықтары қандай?
- PyGame кітапханасы қандай тілде жазылған?
- PyGame кітапханасын қолдану үшін қандай бастапқы білім қажет?
- PyGame қандай ойын түрлерін жасауға мүмкіндік береді?
- PyGame мен басқа ойын жасау кітапханаларының айырмашылығы қандай?
- PyGame кітапханасын орнату үшін қандай команданы қолданамыз?

Практикалық жұмыс 2: PyGame кітапханасының модульдері туралы түсінік. Ойын терезесін жасау үшін PyGame кітапханасының дайын модульдерін қолдану.

Кітапхана - бұл функциялар мен кластардан тұратын кодтар жинағы. Python бағдарламалау тілінде мұндай кодтар жиынтығы модуль деп аталады. PyGame - бұл ойындар жасауда жиі қолданылатын кітапхана, ол да бірнеше ішкі модульдерден тұрады. Мысалы, pygame.draw, pygame.image, pygame.mouse сынды ішкі модульдер арқылы түрлі графикалық элементтерді сызуға, суреттерді жүктеуге және тышқанмен әрекет жасауға болады. PyGame кітапханасының осы дайын модульдерін пайдалана отырып, ойын терезесін оңай құруға мүмкіндік бар.

PyGame-нің негізгі модульдері Модуль атауы Мақсаты pygame.cdrom CD-ROM жетектерін басқару және оған кіру pygame.cursors Курсор бейнелерін жүктеу pygame.display Экранға кіру pygame.draw Фигураларды, сызықтарды және нүктелерді сызу pygame.event Сырткы оқиғаларды басқару pygame.font Жүйелік қаріптерді қолдану pygame.image Кескінді жүктеу және сақтау pygame.key Пернетактада басқан пернені оқу pygame.mixer Дыбыстарды жүктеу және ойнату pygame.mouse Тінтуірмен басқару pygame.movie Кинофильм файлдарын ойнату pygame.music Музыка және аудиолармен жұмыс pygame.rect Төртбұрышты аймақтарды басқару pygame.sndarray Дыбыс деректерін басқару pygame.surface Суреттер мен экранды басқару pygame.transform Суреттерді өзгерту және жылжыту pygame.time Уақыт пен кадр жиілігін басқару •

- Тапсырмалар:
- Берілген модуль атқаратын қызметін жазу.

Модуль Қызметі pygame.font pygame.key pygame.rect pygame.time

- pygame.init() функциясы не үшін қажет екенін түсіндір және оны қолданып, бос терезе жаса. 640x480 өлшемді терезе ашылуы тиіс.
- pygame.Surface нысанын қолданып, экранда қызыл төртбұрыш салып көр. Экранда кез келген жерде қызыл түсті төртбұрыш көрінуі керек.
- pygame.display.set_mode() және pygame.display.set_caption() функцияларын қолданып терезенің атауын өзгерт. Терезеде «Менің алғашқы ойыным» деген атау жазылуы қажет.

- pygame.image.load() арқылы сыртқы суретті жүктеп, экранда көрсет. Сурет экран ортасына шығып тұруы керек.
- blit() әдісі арқылы бірнеше объектіні экран бетіне орналастыр. Екі немесе үш түрлі сурет экранда бір уақытта көрсетілсін.
- pygame.font модулін пайдаланып, экранда өз атыңды шығаратын бағдарлама жаз. Мәтін экранның жоғарғы жағына шығып тұруы қажет.
- Пайдаланушы ESC пернесін басқанда терезе жабылатын бағдарлама жаз. Оқиға нысаны (event) дұрыс өңделуі тиіс.
- Пернетақта арқылы төртбұрышты жоғары-төмен-қолға қозғалатын етіп жаса. KEYDOWN және K_LEFT, K_RIGHT, т.с.с. қолдану керек.
- pygame.time.Clock() көмегімен кадр жиілігін (FPS) 60-қа шекте. Ойын үзіліссіз, бірқалыпты жұмыс істеуі тиіс.
- Ойын басталғанда дыбыс шығатын бағдарлама жаз. pygame.mixer.Sound() немесе pygame.mixer.music қолданылсын.
- Өз ойыңнан шағын 2D сахна ойлап тауып, төмендегі компоненттерді қолдан: Фон суреті Бір қозғалыстағы кейіпкер Бір мәтін Бір дыбыс
  Pygame-нің кемінде 5 түрлі модулін қолдану қажет.

Дескриптор:

- PyGame-нің негізгі модульдері қызметін жазады.
- Терминал ашып, дұрыс команда (pip install pygame) тереді
- Орнату процесін сәтті аяқтады немесе пайда болған қатені сипаттайды
- Орнату нәтижесін скриншотпен немесе жазбаша түрде дәлелдейді

Сұрақтар:

- PyGame кітапханасы не үшін қолданылады?
- PyGame-ды орнату үшін қандай команда қолданылады?
- PyGame кітапханасын импорттау қалай жүзеге асырылады?
- PyGame-да негізгі цикл (main loop) не үшін қажет?
- pygame.display.set_mode() функциясы не істейді?
- pygame.display.set_caption() функциясының қызметі қандай?
- pygame.display.update() пен pygame.display.flip() арасындағы айырмашылық неде?
- PyGame-де оқиғалар қалай өңделеді?
- pygame.event.get() функциясы не үшін қолданылады?
- QUIT оқиғасы не білдіреді және оны қалай қолдануға болады?
- Пернетақта оқиғаларын қалай тексеруге болады?
- Тышқанмен байланысты қандай оқиғалар бар?
- pygame.time.Clock() не үшін қолданылады?
- FPS (frames per second) дегеніміз не және оны қалай орнатамыз?
- Уақытша кідірісті қалай жүзеге асыруға болады?
- pygame.image.load() не үшін қолданылады?
- Суретті экранға қалай шығарамыз?
- Суреттердің масштабын қалай өзгертуге болады?
- pygame.draw.rect() қандай фигура сызады?
- pygame.draw.circle() көмегімен не салуға болады?
- Сызықтарды сызу үшін қандай функция қолданылады?
- pygame.mixer.init() не үшін керек?
- pygame.mixer.Sound() қандай мақсатта қолданылады?
- Дыбысты ойнату үшін қандай әдіс қолданылады?
- pygame.font.Font() не үшін керек?
- Мәтінді экранға шығару процесі қалай жүреді?

- Мәтін түсін, өлшемін қалай өзгерте аламыз?
- Surface дегеніміз не және ол қалай жұмыс істейді?

Практикалық сабақ 3: PyGame терезесі. Суретті жүктеу. Rect нысаны

PyGame кітапханасын қосу. Ойын терезесі: өлшемі, жағдайы

```
gameScreen = pyame.display.set_mode ((400, 300))
```

os модулі - терезенің орналасуы Ойын циклі, ойыннан шығу

```
runGame = True # жалауша ойыннан шығады
```

Оқиғаны бақылау: терезені жабу

```
оқиға үшін pygame.event.get ():
егер event.type == pygame.QUIT: runGame = Қате
Ойыннан шығу: pygame.quit ()
```

Суретті жүктеу. Pygame.image модулі файлдан суретті жүктеуге және Surface типіндегі нысанды қайтаруға мүмкіндік береді.

```
pygame.image.load («файл жолы») #файлдан жаңа кескінді жүктеу
```

Суретті жүктеу (Windows үшін файл жолы) орналастыруды анықтаңыз файлдан жүктелген суретті (myImage) қамтитын Surface нысанын экрандағы сипатталған жерге жүктеу (myRect)

Rect нысаны Pygame тіктөртбұрышты аудандарды сақтау және басқару үшін Rect нысандарын пайдаланады. Rect сол, жоғарғы, ен және биіктік мәндерінің тіркесімінен жасалуы мүмкін. Rect бұрыннан бар немесе «тік» атрибуты бар питон объектілерінен де жасалуы мүмкін.

```
Rect (сол жақ, жоғарғы, иілгіш, биіктігі) -> Rect
Rect ((сол жақта, үстіңгі жақта), (дөңес, биіктік)) -> Rect
```

Rect әдістері

pygame.Rect.copy Түпнұсқа сияқты бірдей өлшемі бар жаңа тіктөртбұрышты қайтарады. pygame.Rect.move Осы алмастыру арқылы жылжытылған жаңа төртбұрышты қайтарады. X және y аргументтері оң немесе теріс санның кез келген мәні болуы мүмкін. pygame.Rect.move_ip

```
Rect.move () әдісі сияқты, бірақ орнында жұмыс істейді.
```

пирог.Рект.инфлат орнында тіктөртбұрыштың өлшемін үлкейту немесе азайту pyame.Rect.inflate_ip орнында тіктөртбұрыштың өлшемін үлкейту немесе азайту пирог.Рект.кламп тіктөртбұрышты басқасының ішіне жылжытады pyame.Rect.clamp_ip тіктөртбұрышты басқасының ішінде орнында жылжытады pygame.Rect.clip тіктөртбұрышты екіншісіне кесіңіз pygame.Rect.union екі тіктөртбұрышты бір-біріне қосады pygame.Rect.union_ip екі тіктөртбұрышты бір орнына, орнына қосады пирамида.Рект.ұзын көптеген төртбұрыштардың одағы pyame.Rect.unionall_ip орнында көптеген төртбұрыштардың бірлестігі пирог.Рект.фит пропорцияны ескере отырып, тіктөртбұрышты өлшемін өзгертіңіз және жылжытыңыз pyame.Rect.normalize теріс өлшемдерін реттеңіз pygame.Rect.контейнерлер бір тіктөртбұрыштың екінші бөлігінде екенін тексеріңіз pyame.Rect.collidepoint нүктенің тіктөртбұрыштың ішінде екенін тексеріңіз pyame.Rect.colliderect егер екі тіктөртбұрыш қиылысса, тексеріңіз пирог.Рект.коллиделист тізімдегі кемінде бір

тіктөртбұрыштың қиылысатынын тексеріңіз пирог.Ректқоллиделисталь тізімдегі барлық төртбұрыштарды қиып алыңыз Pygame.Rect.collideict сөздікте бір тіктөртбұрыштың қиылысатынын тексеріңіз pyame.Rect.collidedictall сөздік қиылысындағы барлық төртбұрыштарды жасаңыз •

- Тапсырмалар:
- PyGame терезесін құру
- PyGame терезесін 800x600 өлшемінде ашу
- Терезе тақырыбын "Менің ойыным" деп атау үлгісі: import pygame pygame.init() screen = pygame.display.set_mode((800, 600)) pygame.display.set_caption("Менің ойыным")
- Суретті жүктеп, экранға шығару
- pygame.image.load() арқылы бір суретті жүктеу
- blit() әдісі арқылы оны экранға шығару
- Сурет терезенің ортасына орналастырылсын
- Rect нысанын қолдану
- Суретке немесе объектіге Rect нысанын беру
- Rect арқылы координаттарды басқарып, объектіні экранда жылжыту (қалауыңызша - пернелермен) Код үзінді мысалы: rect = image.get_rect() rect.x = 100 rect.y = 150 screen.blit(image, rect)
- Объектіні пернетақта арқылы жылжыту (қосымша)
- Көрсеткі пернелермен (↑ ↓ ← →) Rect нысанын жылжыту
- Жаңа координаттармен суретті қайта салып отыру

Дескриптор:

```
•    pygame.display.set_mode() дұрыс қолданады (800x600)
•    pygame.display.set_caption() арқылы тақырыпты дұрыс орнатады
•    pygame.image.load() арқылы суретті сәтті жүктейді
•    screen.blit() әдісімен суретті экранға шығарады
```

- Сурет экранның ортасына шамалас орналастырады
- Суретке get_rect() арқылы Rect байланыстырады
- rect.x, rect.y мәндерімен координатты басқарады
- Экран жаңартылып, қозғалысты дұрыс көрсетеді

Сұрақтар:

```
•    pygame.init() функциясы не үшін қолданылады?
```

- Терезе өлшемін орнату үшін қандай функция қолданылады?
- pygame.display.set_mode((600, 400)) не істейді?
- pygame.display.set_caption("Ойын терезесі") функциясының мақсаты қандай?
- Терезенің фон түсін қалай орнатуға болады?
- screen.fill((0, 255, 0)) коды не істейді?

Практикалық сабақ 4: Оқиғаларды басқару. Оқиға нысаны. Пернетақтамен жұмыс істеуге арналған функциялар.

Оқиға - бұл Pygame бағдарламаның қодынан тыс бір нәрсе болғанын хабарлау. Оқиғалар, мысалы, пернетақтаны, тінтуірді басып, өңделуін күтіп, кезекке қойылады. Pygame.event модуліндегі get функциясы кезекте күткен соңғы оқиғаны қайтарады және оны кезектен алып тастайды.

Оқиға нысаны. Оқиға кезегін өңдеуге арналған модуль

pyame.event.pump Егер сіз өз ойыңызда оқиғаның басқа функцияларын пайдаланбасаңыз, pygame ішкі әрекеттерді басқаруға мүмкіндік беру үшін pygame.event.pump () - ге қоңырау шалыңыз. pygame.event.get кезектен оқиғаларды алады pyame.event.poll кезектен бір оқиғаны алыңыз pygame.event.wait кезектен бір оқиғаны күту пигаме.вент.пек кезектердің белгілі бір түрдегі оқиғаларды күтіп тұрғанын тексеріңіз пигаме.кеңес.қызық барлық оқиғаларды кезектен алып тастаңыз pyame.event.event_name оқиға түріне атау береді. Жол WordCap стилінде. pyame.event.set_blocked кезекте қандай оқиғаларға жол берілмейтіндігін тексереді pygame.event.set_allowed кезекте қандай оқиғалардың рұқсат етілгендігін тексереді pygame.event.get_blocked оқиға түрінің кезектен оқшауланғанын тексеріңіз pyame.event.set_grab кіріс құрылғыларының басқа бағдарламалармен бөлісуін тексереді pyame.event.get_grab бағдарламаның деректерді енгізу құрылғыларында жұмыс істейтінін тексеріңіз pyame.event.post жаңа оқиғаны кезекке қою пирамида.жеңіл оқиғаның жаңа нысанын жасау pygame.event.EventType SDL оқиғасын білдіретін Python нысаны. Оқиғалардың жеке даналары Event функциясын шақыру арқылы жасалады. EventType түріне тікелей қоңырау шалу мүмкін емес. EventType даналары атрибуттарды тағайындауды және жоюды қолдайды.

Pygame оқиға туралы барлық хабарламаларды оқиға кезегі арқылы бақылайды. Осы модульдегі процедуралар осы оқиға кезегін басқаруға көмектеседі. Енгізу кезегі пирамида дисплейінің модуліне қатты тәуелді. Егер дисплей іске қосылмаған болса және бейне режимі орнатылмаған болса, оқиға кезегі жұмыс істемейді. Пернетақтамен жұмыс істеуге арналған функциялар Бұл модульде пернетақтамен жұмыс істеуге арналған функциялар бар.Оқиғалар кезегі pygame.KEYDOWN және pygame.KEYUP оқиғаларын сіз клавиатураның пернелерін басып, босатқан кезде алады. Екі оқиғада пернетақтадағы әр кілтті көрсететін бүтін санды анықтайтын кілт атрибуты бар. Оқиғада қосымша атрибуттар бар: уникод және сканкод. Юникод - енгізілген таңбаға сәйкес келетін жеке таңбалар жолы. Scancode - платформаның арнайы коды. Кілт коды: pressed_keys = pygame.key.get_pressed ()
егер батырманы бассаңыз [K_SPACE]:
Бос орын пернесі басылды () •

- Тапсырмалар:
- Терезені ESC пернесімен жабу
- PyGame терезесін жасау
- Пайдаланушы ESC пернесін басқанда, бағдарлама жабылуы керек Нұсқа: for event in pygame.event.get(): if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE: running = False
- Пернетақта арқылы төртбұрышты басқару
- Экранда бір түсті төртбұрыш болсын
- Бағыттауыш пернелермен (↑ ↓ ← →) оны қозғалту
- Бірнеше пернені өңдеу
- W, A, S, D пернелерімен кейіпкерді (немесе төртбұрышты) қозғалту
- Әр бағытта қозғалғанда экран жаңарып, объекті жылжуы керек
- KEYDOWN және KEYUP айырмашылығын көрсету (қосымша)
- KEYDOWN кезінде түсін өзгерту
- KEYUP кезінде бұрынғы түсіне қайтару

Мысалы: Объекті қызыл түске боялады, перне жіберілсе - қайта көк түске айналады

Дескриптор:

- KEYDOWN оқиғасын дұрыс қолданады
- event.key == pygame.K_ESCAPE арқылы ESC пернесін тексереді
- Жабылу процесін дұрыс ұйымдастырады
- (running = False) K_LEFT, K_RIGHT, K_UP, K_DOWN пернелерін өңдейді
- Әр перне үшін қозғалыс бағытын дұрыс анықтайды
- Перне басқанда объект түсін өзгертеді
- Перне жіберілгенде түсті бұрынғы қалпына келтіреді

Сұрақтар:

- Pygame кітапханасында оқиғалар жүйесі не үшін қажет?
- pygame.event.get() функциясының рөлі қандай? Қашан және қалай қолданылады?

- Оқиға нысаны (event) дегеніміз не және оның қандай атрибуттары бар?
- Пернетақта пернесі басылғанда қандай оқиға түрі тіркеледі? Ал босатылғанда ше?
- event.type және event.key мәндері қалай қолданылады? Мысал келтір.
- Pygame терезесін жабу оқиғасын қалай анықтаймыз және қалай өңдейміз?
- Бағыттауыш пернелермен (↑ ↓ ← →) жұмыс істеу үшін қандай кодтар (константалар) пайдаланылады?
- Пернетақта арқылы кейіпкерді немесе объектіні экранда қозғалту алгоритмін сипаттаңыз.
- Оқиғаларды өңдеу үшін цикл не үшін қажет? Оны қалай дұрыс ұйымдастыру керек?
- Оқиғалармен жұмыс жасауда жиі кездесетін қателіктер мен оларды болдырмау жолдарын атаңыз.

Практикалық жұмыс 5: Python программалау тілінде 2D ойынын құру әдісі.

№1 «ТЕРЕЗЕ» программасының коды

```
import pygame
pygame.init()
screen = pygame.display.set_mode((1000, 600))
fon= pygame.image.load("kartina.bmp")
okno = fon.get_rect(bottomright=(1000, 600))
screen.blit(fon, okno)
pygame.display.update()
```

Нәтижесі:

Тапсырма: Pygame кітапханасын қолданып, қарапайым 2D ойын терезесін жасаңыз. Терезенің өлшемін, атын және түсін орнатыңыз.

Дескриптор:

- Pygame модулін импорттайды
- Белгілі бір өлшемде терезе ашады
- Терезе аты мен фон түсін орнатады
- ESC немесе «қызыл крестикпен» жабады

Сұрақтар:

- Ойын терезесінің өлшемін 1024x768 етіп қалай өзгертуге болады?
- Терезе фонын жасыл түске өзгерту үшін қандай RGB мәнін қолданамыз?
- Терезе атауын "Python 2D Ойыны" деп қалай орнатуға болады?
- ESC пернесін басқанда терезе жабылатын код жолын көрсетіңіз.
- Қателік: Терезе ашылып, бірақ бірден жабылып қалады. Бұл жағдайда не істеу керек? №2 «ТІНТУІРДІҢ КӨМЕГІМЕН КӨК ТҮСПЕН СУРЕТ САЛУ» программасының коды import pygame as pg import sys WHITE = (255, 255, 255) BLUE = (0, 0, 225) sc = pg.display.set_mode((400, 300)) sc.fill(WHITE) pg.display.update() while 1: for i in pg.event.get(): if i.type == pg.QUIT: sys.exit() pressed = pg.mouse.get_pressed() pos = pg.mouse.get_pos() if pressed[0]: pg.draw.circle(sc, BLUE, pos, 5) pg.display.update() pg.time.delay(20)

Нәтижесі:

Тапсырма: Pygame кітапханасын пайдаланып, тінтуірдің сол жақ батырмасын басу арқылы экранда көк түспен сызық немесе нүктелер салу керек.

Дескриптор:

```
    •    import pygame және pygame.init() пайдаланады
    •    set_mode() арқылы терезе өлшемін орнатады
```

- pygame.MOUSEBUTTONDOWN, pygame.MOUSEMOTION пайдаланады
- pygame.draw.circle() немесе draw.line() арқылы салады

- pygame.display.update() қолданады

Сұрақтар:

- Тінтуірдің оң жақ батырмасын қолдану арқылы басқа түспен салу үшін не өзгерту қажет?
- Қазіргі кодта салынған суреттерді тазалау (clear) мүмкіндігі жоқ. Оны қалай қосуға болады?
- Көк түстің орнына қызыл түс қолдану үшін қандай код жазу керек?
- Нүкте орнына сызық салу үшін қандай Pygame функциясын пайдаланасыз?
- Тінтуірмен салынған суретті файл ретінде сақтау мүмкін бе? Қалай?

№3 «КЛАВИАТУРАМЕН ДӨҢГЕЛЕК САЛУ» программасының коды

```
import pygame as pg
pg.init()
WHITE = (255, 255, 255)
RED = (225, 0, 50)
GREEN = (0, 225, 0)
BLUE = (0, 0, 225)
sc = pg.display.set_mode((400, 300))
running=True
```

while running:

```
    for i in pg.event.get():
        if i.type == pg.QUIT:
            running = False
        elif i.type == pg.KEYDOWN:
            if i.key == pg.K_r:
       pg.draw.circle(sc, RED, (100, 100), 20)
                pg.display.update()
            elif i.key == pg.K_g:
                pg.draw.circle(sc, GREEN, (150, 150), 20)
                pg.display.update()
            elif i.key == pg.K_b:
          pg.draw.circle(sc, BLUE, (200, 100), 20)
                pg.display.update()
            else:
        pg.draw.circle(sc, WHITE, (100, 200), 20)
                pg.display.update()
pg.quit()
```

Нәтижесі:

Тапсырма: Pygame кітапханасын пайдаланып, клавиатура арқылы экранға дөңгелек (шеңбер) салатын бағдарлама құрыңыз. W, A, S, D пернелері арқылы дөңгелекті қозғалту немесе жаңа дөңгелек салу функцияларын жүзеге асырыңыз.

Дескриптор:

```
    •    pygame.init() және import pygame қолданады
    •    pygame.display.set_mode() қолданады
```

- KEYDOWN, KEYUP арқылы пернелерді бақылайды
- pygame.draw.circle() арқылы экранға дөңгелек салады
- W (жоғары), A (солға), S (төмен), D (оңға) бойынша қозғайды

Сұрақтар:

- Дөңгелекті тек бір рет салып, оны тек қозғалтқыңыз келсе, кодты қалай өзгертер едіңіз?
- Дөңгелектің түсін көкке (BLUE) ауыстыру үшін не өзгерту қажет?

- Егер дөңгелек экраннан шығып кетсе, оны шектеп қою үшін не істеуге болады?
- Қосымша перне (мысалы, SPACE) басылғанда экранды тазалау үшін қандай код жазасыз?
- Әр перне басқан сайын жаңа дөңгелек салудың орнына, бір дөңгелектің орнын ауыстыруға қалай болады?

№4 «ТІНТУІРДІ БАСУ АРҚЫЛЫ ДӨҢГЕЛЕК АЛУ» программасының коды

```
import pygame as pg                                                                    import sys
WHITE = (255, 255, 255)
RED = (225, 0, 50)
GREEN = (0, 225, 0)
BLUE = (0, 0, 225)
sc = pg.display.set_mode((400, 300))
sc.fill(WHITE)
pg.display.update()
```

while 1:

```
    for i in pg.event.get():
        if i.type == pg.QUIT:
            sys.exit()
        if i.type == pg.MOUSEBUTTONDOWN:
            if i.button == 1:
```

pg.draw.circle(sc, RED, i.pos, 20)

```
                pg.display.update()
            elif i.button == 3:
```

pg.draw.circle(sc, BLUE, i.pos, 20) pg.draw.rect(sc, GREEN,

```
                    (i.pos[0] - 10,
                     i.pos[1] - 10,
                    20, 20))
                pg.display.update()
            elif i.button == 2:
                sc.fill(WHITE)
                pg.display.update()
    pg.time.delay(20)
```

Нәтижесі:

Тапсырма: Pygame кітапханасын пайдаланып, тінтуірдің сол жақ батырмасын басқан кезде экранда дөңгелек (шеңбер) шығатындай бағдарлама құрыңыз. Дескриптор:

```
•    import pygame және pygame.init() қолданады
•    pygame.display.set_mode() арқылы терезе құра алады
```

- MOUSEBUTTONDOWN оқиғасын өңдей алады
- pygame.draw.circle() арқылы экранға дөңгелек шығарады
- Оқиғаларды тексеріп, pygame.display.update() қолданады

Сұрақтар:

```
    •    әр басқанда дөңгелек орнына төртбұрыш (шаршы) шығару үшін не өзгерту керек?
```

- Дөңгелектің радиусын 50 пиксельге арттыру үшін қай жерін өзгерту керек?
- Дөңгелек түсін көк қылу үшін қандай RGB мәнін қолдану керек?
- Тек оң жақ тінтуір батырмасымен салу үшін қандай шарт жазу керек?
- Экранда 5 дөңгелектен артық болмаса, қалай шектеу қоюға болады?

№5 «ШАРДЫҢ ДИАГОНАЛ БОЙЫНША ҚОЗҒАЛЫСЫ» программасының кодын жазу.

```
import pygame
```

pygame.init() size=[600,600]

```
screen = pygame.display.set_mode(size)
screen.fill([255,255,255])
pygame.display.flip()
x=20; y=20
flag = True
```

Used to manage how fast the screen updates

```
clock = pygame.time.Clock()
```

while flag:

```
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            flag = False
    pygame.draw.circle(screen, [0,0,255], (x,y),40)
    y=y+20;x=x+20
    pygame.display.flip()
    screen.fill([255,255,255])
    clock.tick(10)
pygame.quit()
```

Нәтижесі:

Тапсырма: Pygame кітапханасын пайдалана отырып, экранда шар (дөңгелек) диагональ бағытта (оңға-төмен немесе солға-жоғары) үздіксіз қозғалатындай бағдарлама жазыңыз. Шар экран шетіне жеткенде шекарадан шығып кетпей, бағытын өзгертіңіз.

Дескриптор:

- pygame модулін қосып, терезе инициализация жасайды
- pygame.draw.circle() арқылы дөңгелекті экранға салады
- x және y координаталарын бір уақытта өзгерте алады
- Шар экран шетіне жеткен кезде қозғалыс бағытын кері аударады (reverse)
- while, clock.tick(), pygame.display.update() қолданылады

Сұрақтар:

- Шар қозғалыс жылдамдығын 10 пиксель/кадр ету үшін кодта не өзгерту керек?
- Шардың түсін жасылға өзгерту үшін RGB кодын қалай жазасыз?
- Шар экран шетінен шықпай тоқтап қалу үшін қандай шарт енгізу қажет?

- Егер диагональ қозғалыс орнына тек көлденең (оң-сол) қозғалыс керек болса, қай код бөлігін өзгерту керек?
- Диагональ бағытты өзгерту үшін қандай математикалық логика қолданылады?
- Pygame-де уақытты басқару үшін қандай функциялар қолданылады?

Практикалық жұмыс 6: Python программалау тілінде 2D ойынын құру әдісі

№1 «КУРСОРДЫ ЖЫЛЖЫТУ» программасының коды

```
import pygame as pg
import sys
WHITE = (255, 255, 255)
BLUE = (0, 0, 225)
pg.init()
sc = pg.display.set_mode((400, 300))
sc.fill(WHITE)
pg.display.update()
pg.mouse.set_visible(False)
```

while 1:

```
    for i in pg.event.get():
        if i.type == pg.QUIT:
            sys.exit()
    sc.fill([0,0,0])
    if pg.mouse.get_focused():
        pos = pg.mouse.get_pos()
```

pg.draw.circle(

```
            sc, BLUE, (pos[0] - 10,
                       pos[1] - 10), 25)
    pg.display.update()
    pg.time.delay(20) import pygame as pg
import sys
WHITE = (255, 255, 255)
BLUE = (0, 0, 225)
pg.init()
sc = pg.display.set_mode((400, 300))
sc.fill(WHITE)
pg.display.update()
pg.mouse.set_visible(False)
```

while 1:

```
    for i in pg.event.get():
        if i.type == pg.QUIT:
            sys.exit()
    sc.fill([0,0,0])
    if pg.mouse.get_focused():
        pos = pg.mouse.get_pos()
```

pg.draw.circle(

```
            sc, BLUE, (pos[0] - 10,
                       pos[1] - 10), 25)
    pg.display.update()
    pg.time.delay(20)
```

Нәтижесі:

Тапсырма: Төмендегі мәтінді теріп жазыңыз. Мәтінді теру барысында курсорды тек бағыттау пернелерімен (← ↑ → ↓) жылжытыңыз.

- Мен" сөзін тауып, курсорды сол сөздің соңына апарып, " бүгін" сөзін қосыңыз.
- "жылжыту" сөзінен кейін үтір қойыңыз.
- Соңғы сөйлемнің соңына "!" белгісін қойыңыз.

Дескриптор:

- Мәтінді толық және қатесіз тереді
- Курсорды тек бағыттау пернелерімен басқарады
- Сөз арасына қосымша сөзді дұрыс қосады
- Белгіленген тыныс белгілерін дұрыс қояды

Сұрақтар:

- Курсор дегеніміз не?
- Курсорды жылжыту үшін қандай пернелер қолданылады?
- Курсорды бір сөйлемнің басына қалай апаруға болады?
- Enter пернесінің курсорға әсері қандай?
- Backspace пен Delete пернелері курсордың қай жағындағы мәтінді өшіреді?

№2 «ТӨРТБҰРЫШТЫ ШЫҒАРУ» программасының коды

```
import pygame
BLACK = ( 0, 0, 0)
#WHITE = (255, 255, 255)
YELLOW   = (255, 255, 0)
FPS = 30
pygame.init()
screen = pygame.display.set_mode((500, 400))
#screen_rect = screen.get_rect()
pygame.display.set_caption("rectangle")
rectangle = pygame.rect.Rect(20, 20, 70, 70)
rectangle_draging = False
clock = pygame.time.Clock()
running = True
```

while running:

```
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.MOUSEBUTTONDOWN:
            if event.button == 1:
                if rectangle.collidepoint(event.pos):
                    rectangle_draging = True
```

mouse_x, mouse_y = event.pos

```
                    offset_x = rectangle.x - mouse_x
                    offset_y = rectangle.y - mouse_y
        elif event.type == pygame.MOUSEBUTTONUP:
            if event.button == 1:
                rectangle_draging = False
        elif event.type == pygame.MOUSEMOTION:
```

if rectangle_draging: mouse_x, mouse_y = event.pos rectangle.x = mouse_x + offset_x rectangle.y = mouse_y + offset_y

```
    screen.fill(BLACK)
    pygame.draw.rect(screen, YELLOW, rectangle)
    pygame.display.flip()
    clock.tick(FPS)
pygame.quit()
import pygame
BLACK = ( 0, 0, 0)
#WHITE = (255, 255, 255)
YELLOW   = (255, 255, 0)
FPS = 30
pygame.init()
screen = pygame.display.set_mode((500, 400))
#screen_rect = screen.get_rect()
pygame.display.set_caption("rectangle")
rectangle = pygame.rect.Rect(20, 20, 70, 70)
rectangle_draging = False
clock = pygame.time.Clock()
running = True
```

while running:

```
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.MOUSEBUTTONDOWN:
            if event.button == 1:
                if rectangle.collidepoint(event.pos):
                    rectangle_draging = True
```

mouse_x, mouse_y = event.pos

```
                    offset_x = rectangle.x - mouse_x
                    offset_y = rectangle.y - mouse_y
        elif event.type == pygame.MOUSEBUTTONUP:
            if event.button == 1:
        rectangle_draging = False
        elif event.type == pygame.MOUSEMOTION:
```

if rectangle_draging: mouse_x, mouse_y = event.pos rectangle.x = mouse_x + offset_x rectangle.y = mouse_y + offset_y

```
    screen.fill(BLACK)
    pygame.draw.rect(screen, YELLOW, rectangle)
    pygame.display.flip()
    clock.tick(FPS)
pygame.quit()
```

Нәтижесі:

Тапсырма:

---

1 МЫНА КОДТЫ ЖАЗ ЖӘНЕ ОНЫҢ НЕ ІСТЕЙТІНІН ТҮСІНДІРІҢІЗ: PYTHON

КопироватьРедактировать

```
for i in range(5):
    print("*" * 10)
```

- Кодты орында.
- Нәтижесін сипатта.
- Төртбұрыштың биіктігі мен енін өзгертіп көр.

---

2 ТӨРТБҰРЫШТЫ САЛЫҢЫЗ (PYTHON TURTLE) ТӨМЕНДЕГІ ӘРЕКЕТТЕРДІ PYTHON АРҚЫЛЫ ОРЫНДАҢЫЗ:

- turtle модулін қос.
- Қаламды түсір.
- 100 қадам алға жүр.
- 90 градусқа оңға бұрыл.
- Бұл әрекеттерді 4 рет қайтала.
- Экранда толық төртбұрыш пайда болуын қамтамасыз ет.

Дескриптор:

- Төртбұрышты экранға шығару алгоритмін жазады
- Ені мен биіктігін өзгертіп эксперимент жасайды
- Қайталау (цикл) операторын дұрыс қолданады
- Төртбұрыш формасын дұрыс түсіндіреді

Сұрақтар:

- Төртбұрыш дегеніміз не? Оның қандай қасиеттері бар?
- Программада цикл не үшін қолданылады?
- Қай команда төртбұрыштың бұрыштарын жасауға көмектеседі?
- Егер төртбұрыштың ені мен биіктігін тең жасасақ, не болады?
- Төртбұрышты шығару үшін қандай пернелер мен командалар жиі қолданылады?

№3 «Диагонал бойынша дөңгелектің қозғалысы және кері жүру» программасының коды

```
 import pygame
pygame.init()
screen = pygame.display.set_mode((600,600))
x=0; y=0
running = True
clock = pygame.time.Clock()
flag=0
```

while running:

```
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    if flag==0:
        pygame.display.flip()
        screen.fill([0, 0, 0])
        pygame.draw.circle(screen, (0,255,0),[x,y],20,5)
        x=x+10;y=y+10
        clock.tick(10)
    elif flag==1:
```

```
        pygame.display.flip()
        screen.fill([0, 0, 0])
        pygame.draw.circle(screen, (0,255,0),[x,y],20)
        x=x-10; y=y-10
        clock.tick(10)
    if x>=580 and y>=580:
        flag=1
    elif x<20 and y<20:
        flag=0
pygame.quit()
```

Нәтижесі:

Тапсырмалар:

- Дөңгелек (шеңбер пішініндегі объект) жасап, оны экранның бір бұрышына орналастырыңыз.
- Ол диагональ бойынша (оңға және төменге) қозғала бастасын.
- Экран шетіне жеткенде кері бағытта (солға және жоғарыға) қозғалуын қамтамасыз ет.
- Бос орын пернесін басқанда қозғалыс тоқтап, қайта басқанда жалғассын.
- Диагональ қозғалысқа түс қосу (түсі өзгерсін).
- Қозғалып келе жатып із қалдыру.
- Диагональ траекторияны өзгертіп, зигзаг салдыру.

Дескриптор:

- Объектіні экранда шеңбер түрінде жасайды
- Объектіні диагональ бағытта қозғалтады
- Объект экран шетіне жеткенде бағытын өзгертеді
- Бос орын пернесі арқылы қозғалысты тоқтатады/жалғастырады

Сұрақтар:

- Диагональ бойынша қозғалу үшін X және Y координаталары қалай өзгеруі керек?
- Объект экран шетіне жеткенін қалай анықтауға болады?
- dx = -dx деген не үшін қолданылады?
- Қозғалысты тоқтатып, қайта қосу үшін қандай оқиға (event) қажет?
- Қозғалыс жылдамдығын арттыру немесе баяулату үшін не істеу керек? №4 «Кездейсоқ шарлардың әр түрлі түске боялу» программасының коды import sys import pygame import random pygame.init() screen = pygame.display.set_mode((640, 480)) clock = pygame.time.Clock() while True: for event in pygame.event.get(): if event.type == pygame.QUIT: pygame.quit() sys.exit() if event.type == pygame.KEYDOWN: r=random.randint(0,255) g=random.randint(0,255) b=random.randint(0,255) color = (r, g, b) x=random.randint(50,590) y=random.randint(50,430) pygame.draw.circle(screen, color, (x,y), 50) pygame.display.flip() clock.tick(10) pygame.draw.circle(screen, (0,0,0), (x,y), 50) Нәтижесі:

Тапсырма:

```
    •    Экранда кемінде 10 дөңгелек (шар) шығарыңыз.
```

- Әр шар экранда кездейсоқ орынға орналасуы керек.
- Әр шар кездейсоқ түске боялуы керек.
- Барлық шарлар бірдей өлшемде болсын.
- Әр шардың өлшемін де кездейсоқ етіп өзгерту.
- Шардың орны мен түсі ғана емес, шеңбердің сызық түсі (pen.color) мен қалыңдығы (pensize) өзгеретіндей ету.
- Шарлар жолмен (мысалы, диагональ) салынатын болсын.

Дескриптор:

- random және turtle модульдерін қосады
- Дөңгелектерді кездейсоқ координаталарда орналастырады
- Әр дөңгелекті әр түрлі түспен бояйды
- 10 шар салынды және олардың өлшемі бірдей болды

Сұрақтар:

```
    •    random.randint(-200, 200) функциясы не істейді?
    •    random.choice(colors) не үшін қолданылады?
    •    pen.begin_fill() және pen.end_fill() командаларының рөлі қандай?
```

- Барлық шарды бірдей өлшемде ету үшін қандай параметр тұрақты болуы керек?
- Шардың түсін бірдей етіп жасау үшін не өзгерту қажет?

Практикалық жұмыс 7: Tkinter кітапханасы

Python тілінде ішкі GUI құру құралдары қарастырылмаған. Графикалық интерфейсі бар қарапайым қосымшалар жасау үшін көбіне tkinter модулі қолданылады. Бұл модуль арқылы түрлі графикалық элементтері бар шағын бағдарламалар жазуға мүмкіндік беріледі. "Tkinter" атауы "Tk interface" (Tk интерфейсі) тіркесінен шыққан. Бұл кітапхана графикалық компоненттерді, яғни виджеттерді, соның ішінде canvas ("кенеп") атты элементті қолдана отырып, векторлық графикалар салуға арналған. Tkinter көмегімен әртүрлі геометриялық пішіндерді - қисықтар, доғалар, эллипстер, тіктөртбұрыштар мен түзу сызықтарды бейнелеуге болады. Сонымен қатар, координаттары белгілі бір формулалармен есептелетін графиктерді, мысалы, тригонометриялық функциялардың графиктерін құру мүмкіндігі бар. Алайда, tkinter модулін қолданатын кейбір бағдарламалар IDLE ортасында тұрақсыз жұмыс істеуі мүмкін, себебі IDLE-дің өз ішінде де tkinter пайдаланылады. Сондықтан мұндай бағдарламаларды операциялық жүйенің командалық жолы арқылы іске қосу - анағұрлым тиімді шешім.

№1 Tkinter модулін пайдаланып экранға бос терезе шығару программасының коды

```
Программа 13.1 (empty_window1.py)
```

Бұл программа экранға бос терезе шығарады.

```
import tkinter       # tkinter модулін импорттайды.
 def main():
```

Басты терезе интерфейсінің элементін құру.

```
main_window = tkinter.Tk()
```

tkinter ортасының басты цикліне кіру

```
tkinter.mainloop()
 main () # Басты терезені шақыру.
```

Нәтижесі:

Тапсырма:

- 1 Қарапайым терезе жасау
- Tkinter көмегімен терезе ашыңыз.
- Терезеге тақырып қой (мысалы, "Менің бірінші терезем").
- Терезе өлшемін орнатыңыз (мысалы: 400x300).

- Бір батырма (Button) орналастырыңыз.
- Батырма басылғанда, консольға "Сәлем, әлем!" деген мәтін шықсын. •
- 2 Қолданушыдан мәлімет алу
- Терезеге мәтін енгізу өрісін (Entry) және батырма орналастырыңыз.
- Қолданушы өз атын енгізеді.
- Батырма басылғанда экранда (Label арқылы) «Сәлем, [аты]!» деп шықсын.

Дескриптор

- Tkinter кітапханасында программа құрады.
- Терезе (Tk()) құрады және тақырып орнатады
- Батырма (Button) және мәтінді енгізу өрісін (Entry) қолданады
- Функция арқылы оқиғаны өңдейді (батырма басу)
- Label көмегімен мәтінді экранға шығарады

Сұрақтар:

```
•    tk.Tk() не үшін қолданылады?
```

- command= параметрі не істейді?
- entry.get() функциясы қандай мән қайтарады?
- pack() және place() командаларының айырмашылығы неде?
- Қолданушы әрекетіне жауап беретін функция қалай жазылады?

Практикалық жұмыс 8: Tkinter кітапханасының командалары. Tkinter кітапханасының негізгі модульдері

Tkinter модулі интерфейстің 15 элементін қолданады. Біз олардың бәрін қарастыра алмаймыз, дегенмен енгізілген мәліметтерді жинақтап экранға шығаратын қарапайым программа құру тәсілдерін қарастырамыз. Tkinter интерфейсі элементтері tkinter модулін пайдалану

Элемент Сипаттамасы Button

```
Басқан (шерткен) кезде бір әрекет орындайтын батырма
```

Canvas График сызып көрсету үшін қолданылатын тіктөртбұрышты аймақ Checkbutton

```
"іске қосылған" (on) немесе "қосылмаған" (off) деген екі жағдайды көрсететін батырма
```

Entry Қолданушының пернетақтадан бір жол мәлімет енгізуіне

```
арналған мәтіндік өріс (аймақ)
```

Scrollbar Жылжу тиегін жоғары-төмен қозғалту үшін (скроллинг), интерфейс элементтерінің кейбір типтерін пайдалану кезінде қолданыла алады Text Мәтін енгізу кезінде оның көптеген жолдарын енгізу мүмкіндігін беретін интерфейс элементі Toplevel Frame интерфейсі элементіне ұқсас контейнер, одан айырмасы экранға өз терезесі шығарылады Frame Интерфейстің басқа элементін сақтай алатын контейнер Label Экранға бір жол мәтін немесе суреттік бейне шығаратын аймақ Listbox Ішінен мән таңдап алуға болатын тізім Menu Қолданушы Menubutton интерфейсі элементін шерткен кезде, экранға шығатын меню пункттерінің тізімі Menubutton Экранға шығарылатын және қолданушы тышқанмен оны шертуге болатын меню Message Көптеген жолдары бар мәтін шығарады Radiobutton Таңдап алуға немесе таңдамауға болатын интерфейс элементі. Radiobutton интерфейсінің элементтері көбінесе топтарда болады және ондағы бірнеше мүмкіндіктердің біреуін таңдап алуға болады Scale Қолданушыға жылжу тиегін (ползунок - жылжу тиегі, жылжымақ) шкала бойынша сырғыта отырып, мән таңдап алуға мүмкіндік беретін интерфейс элементі

№1 Tkinter модулін пайдаланып экранға бос терезе шығару программасының коды

```
# Программа 13.2 (empty_window2.py)
```

Бұл программа бос терезе көрсетеді.

```
import tkinter
class MyGUI:
def __init__(self) :
```

Басты терезе интерфейсінің элементін құру.

```
self.main_window = tkinter.Tk()
```

tkinter ортасының басты цикліне кіру.

Экранға бос терезе шығару

```
tkinter.mainloop()
```

MyGUI класының экземплярын жасау

```
 my _gui = MyGUI ()
```

Нәтижесі: • • • • • • • • •

- Тапсырма:
- 1 Қарапайым терезе жасау
- Tkinter көмегімен терезе аш.
- Терезеге тақырып қой (мысалы, "Менің бірінші терезем").
- Терезе өлшемін орнат (мысалы: 400x300).
- Бір батырма (Button) орналастыр.
- Батырма басылғанда, консольға "Сәлем, әлем!" деген мәтін шықсын.
- 2 Қолданушыдан мәлімет алу
- Терезеге мәтін енгізу өрісін (Entry) және батырма орналастыр.
- Қолданушы өз атын енгізеді.
- Батырма басылғанда экранда (Label арқылы) «Сәлем, [аты]!» деп шықсын.

Дескриптор:

- Tkinter кітапханасында программа құрады.
- Терезе (Tk()) құрады және тақырып орнатады
- Батырма (Button) және мәтін енгізу өрісін (Entry) қолданады
- Функция арқылы оқиғаны өңдейді (батырма басу)
- Label көмегімен мәтінді экранға шығарады

Сұрақтар:

```
    •    tk.Tk() не үшін қолданылады?
```

- command= параметрі не істейді?

- entry.get() функциясы қандай мән қайтарады?
- pack() және place() командаларының айырмашылығы неде?
- Қолданушы әрекетіне жауап беретін функция қалай жазылады?
- Tkinter терезесін ашу үшін қандай негізгі командалар қолданылады?
- mainloop() функциясының рөлі қандай? Онсыз не болады?
- Интерфейске бір уақытта бірнеше батырма орналастыру үшін не істеу керек?
- Мәтін енгізу өрісі (Entry) арқылы қолданушыдан ақпаратты қалай алуға болады?
- Label мен Button элементтері арасындағы айырмашылық неде?

№2 «Жылан» программасының коды

```
import pygame
import random
```

initialize pygame

```
pygame.init()
```

create display & run update

```
width = 640
height = 480
display = pygame.display.set_mode((width, height))
pygame.display.update()
pygame.display.set_caption("Snakee game")
```

define colors

```
colors = {
    "snake_head": (0, 255, 0),
"snake_tail": (0, 200, 0),
    "apple": (255, 0, 0)
```

} # snake position with offsets

```
snake_pos = {
```

"x": width/2-10, "y": height/2-10, "x_change": 0, "y_change": 0 } # snake el size

```
snake_size = (10, 10)
```

current snake movement speed

```
snake_speed = 5
```

snake tails

```
snake_tails = []
snake_pos["x_change"] = -snake_speed
```

```
for i in range(75):
    snake_tails.append([snake_pos["x"] + 10*i, snake_pos["y"]])
```

food

```
food_pos = {
    "x": round(random.randrange(0, width - snake_size[0]) / 10) * 10,
    "y": round(random.randrange(0, height - snake_size[1]) / 10) * 10,
```

}

```
food_size = (10, 10)
food_eaten = 0
```

start loop

```
game_end = False
clock = pygame.time.Clock()
```

while not game_end: # game loop

```
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_end = True
        elif event.type == pygame.KEYDOWN:
      if event.key == pygame.K_LEFT and snake_pos["x_change"] == 0:
```

move left

```
                snake_pos["x_change"] = -snake_speed
                snake_pos["y_change"] = 0
elif event.key == pygame.K_RIGHT and snake_pos["x_change"] == 0:
```

move right

```
                snake_pos["x_change"] = snake_speed
                snake_pos["y_change"] = 0
      elif event.key == pygame.K_UP and snake_pos["y_change"] == 0:
```

move up

```
                snake_pos["x_change"] = 0
                snake_pos["y_change"] = -snake_speed
elif event.key == pygame.K_DOWN and snake_pos["y_change"] == 0:
```

move down

```
                snake_pos["x_change"] = 0
                snake_pos["y_change"] = snake_speed
```

clear screen

```
    display.fill((0,0,0))
```

move snake tails

```
    ltx = snake_pos["x"]
    lty = snake_pos["y"]
    for i,v in enumerate(snake_tails):
        _ltx = snake_tails[i][0]
        _lty = snake_tails[i][1]
        snake_tails[i][0] = ltx
        snake_tails[i][1] = lty
        ltx = _ltx
        lty = _lty
```

draw snake tails for t in snake_tails:

```
        pygame.draw.rect(display, colors["snake_tail"], [
            t[0],   t[1],
            snake_size[0],
            snake_size[1]])
```

draw snake

```
    snake_pos["x"] += snake_pos["x_change"]
    snake_pos["y"] += snake_pos["y_change"]
```

teleport snake, if required

```
    if(snake_pos["x"] < -snake_size[0]):
        snake_pos["x"] = width
    elif(snake_pos["x"] > width):
        snake_pos["x"] = 0
    elif(snake_pos["y"] < -snake_size[1]):
        snake_pos["y"] = height
    elif(snake_pos["y"] > height):
        snake_pos["y"] = 0
    pygame.draw.rect(display, colors["snake_head"], [
        snake_pos["x"],
        snake_pos["y"],
        snake_size[0],
        snake_size[1]])
```

draw food

```
    pygame.draw.rect(display, colors["apple"], [
        food_pos["x"],
        food_pos["y"],
        food_size[0],
        food_size[1]])
```

detect collision with food

```
    if(snake_pos["x"] == food_pos["x"]
        and snake_pos["y"] == food_pos["y"]):
```

food_eaten += 1

```
        snake_tails.append([food_pos["x"], food_pos["y"]])
        food_pos = {
      "x": round(random.randrange(0, width - snake_size[0]) / 10) * 10,
      "y": round(random.randrange(0, height - snake_size[1]) / 10) * 10, }
```

detect collision with tail

```
    for i,v in enumerate(snake_tails):
        if(snake_pos["x"]+snake_pos["x_change"] == snake_tails[i][0]
            and snake_pos["y"]+snake_pos["y_change"] == snake_tails[i][1]):
            snake_tails = snake_tails[:i]
```

break

```
    pygame.display.update()
```

set FPS

```
    clock.tick(30)
```

close app, if required

```
pygame.quit()                quit()
```

Нәтижесі:

Тапсырма:

- Қарапайым жылан қозғалысы
- Экран ашып, жыланның басын (квадрат не шеңбер) жасаңыз.
- Жылан жоғары, төмен, солға, оңға бағытта қозғалсын.
- Пернетақта арқылы басқару мүмкіндігін қосыңыз.
- Шекарадан шыққан кезде ойын тоқтасын немесе басталу нүктесіне оралсын.

Дескриптор:

- Экран мен жылан нысанын дұрыс жасайды
- Жылан төрт бағытта қозғалатындай код жазады
- Пернетақтамен басқару функцияларын дұрыс қолданады
- Шартты логикамен қозғалысты шектеді (кері бағытқа бұрылмау)

Сұрақтар:

```
    •    turtle.Turtle() не үшін қолданылады?
```

- Жылан бағытын өзгерту үшін қандай функциялар жазылды?
- Неліктен бір бағытта қозғалып жатқанда кері бұрылуға болмайды?
- penup() және pendown() арасындағы айырмашылық қандай?
- Жылан экран шетіне жеткенде не болады? Оны қалай анықтауға болады?
- screen.onkey() командасы не үшін қажет?

- Егер жыланның жылдамдығын арттырғыңыз келсе, қандай жолдар бар?
- Жылан өз денесіне соғылғанын қалай тексеруге болады?
- Жылан жем жесе, оны қалай ұзарту керек?
- Ойын соңын (game over) қалай ұйымдастырар едің?

№3 «САПЁР» программасының коды #!/usr/bin/env python

```
__author__ = "Dmitriy Krasota aka g0t0wasd"
```

Minesweeper in Python using Tkinter. # More at http://pythonicway.com/python-games/python-arcade/31-python-minesweep

```
from tkinter import *
import random
GRID_SIZE = 20 #   өріс өлшемі
SQUARE_SIZE = 20 # клетка өлшемі
MINES_NUM = 40 #  өрістегі мина саны
mines = set(random.sample(range(1, GRID_SIZE**2+1), MINES_NUM))
```

Минаны өріске кездейсоқ орнату

```
clicked = set()
```

Біз басқан барлық жасушаларды сақтайтын жиынтық

```
def check_mines(neighbors):
```

""" Көршілер айналасындағы миналар санын қайтаратын Функция"neighbors """

```
    return len(mines.intersection(neighbors))
def generate_neighbors(square):
```

""" Square-ге іргелес ұяшықтарды қайтарады """

```
    if square == 1:
        data = {GRID_SIZE + 1, 2, GRID_SIZE + 2}
    elif square == GRID_SIZE ** 2:
        data = {square - GRID_SIZE, square - 1, square - GRID_SIZE - 1}
        elif square == GRID_SIZE:
        data = {GRID_SIZE - 1, GRID_SIZE * 2, GRID_SIZE * 2 - 1}
    elif square == GRID_SIZE ** 2 - GRID_SIZE + 1:
        data = {square + 1, square - GRID_SIZE, square - GRID_SIZE + 1}
        elif square < GRID_SIZE:
        data = {square + 1, square - 1, square + GRID_SIZE,
```

square + GRID_SIZE - 1, square + GRID_SIZE + 1}

```
    elif square > GRID_SIZE ** 2 - GRID_SIZE:
        data = {square + 1, square - 1, square - GRID_SIZE,
```

square - GRID_SIZE - 1, square - GRID_SIZE + 1}

```
    elif square % GRID_SIZE == 0:
        data = {square + GRID_SIZE, square - GRID_SIZE, square - 1,
```

square + GRID_SIZE - 1, square - GRID_SIZE - 1}

```
    elif square % GRID_SIZE == 1:
        data = {square + GRID_SIZE, square - GRID_SIZE, square + 1,
```

square + GRID_SIZE + 1, square - GRID_SIZE + 1}

```
        else:
        data = {square - 1, square + 1, square - GRID_SIZE, square + GRID_SIZE,
```

square - GRID_SIZE - 1, square - GRID_SIZE + 1, square + GRID_SIZE + 1, square + GRID_SIZE - 1}

```
    return data
def clearance(ids):
    clicked.add(ids) # добавляем нажатую клетку в сет нажатых
    ids_neigh = generate_neighbors(ids)
    around = check_mines(ids_neigh)
    c.itemconfig(ids, fill="green")
    if around == 0:
        neigh_list = list(ids_neigh)
                while len(neigh_list) > 0:
            item = neigh_list.pop()
                        c.itemconfig(item, fill="green")
            item_neigh = generate_neighbors(item)
            item_around = check_mines(item_neigh)
```

if item_around > 0: if item not in clicked:

```
                    x1, y1, x2, y2 = c.coords(item)
```

c.create_text(x1 + SQUARE_SIZE / 2, y1 + SQUARE_SIZE / 2,

```
                                  text=str(item_around),
                                  font="Arial {}".format(int(SQUARE_SIZE / 2)),
                                  fill='yellow')
                        else:
                                neigh_list.extend(set(item_neigh).difference(clicked))
                                neigh_list = list(set(neigh_list))
            clicked.add(item)
    else:
        x1, y1, x2, y2 = c.coords(ids)
```

c.create_text(x1 + SQUARE_SIZE / 2, y1 + SQUARE_SIZE / 2,

```
text=str(around),
                      font="Arial {}".format(int(SQUARE_SIZE / 2)),
                      fill='yellow')
def rec_clearance(ids):
  clicked.add(ids)
    neighbors = generate_neighbors(ids)
    around = check_mines(neighbors)
```

if around:

```
        x1, y1, x2, y2 = c.coords(ids)
        c.itemconfig(ids, fill="green")
```

c.create_text(x1 + SQUARE_SIZE / 2, y1 + SQUARE_SIZE / 2,

```
                    text=str(around),
                    font="Arial {}".format(int(SQUARE_SIZE / 2)),
                    fill='yellow')
    else:
        for item in set(neighbors).difference(clicked):
            c.itemconfig(item, fill="green")
            rec_clearance(item)

def click(event):
    ids = c.find_withtag(CURRENT)[0]
```

if ids in mines:

```
        c.itemconfig(CURRENT, fill="red")
    elif ids not in clicked:
        clearance(ids)
        c.itemconfig(CURRENT, fill="green")
    c.update()
def mark_mine(event):
    ids = c.find_withtag(CURRENT)[0]
```

if ids not in clicked:

```
        clicked.add(ids)
        x1, y1, x2, y2 = c.coords(ids)
        c.itemconfig(CURRENT, fill="yellow")
    else:
        clicked.remove(ids)
        c.itemconfig(CURRENT, fill="gray")
root = Tk()
root.title("Pythonicway Minesweep")
c = Canvas(root, width=GRID_SIZE * SQUARE_SIZE, height=GRID_SIZE * SQUARE_SIZE)
c.bind("<Button-1>", click)
c.bind("<Button-3>", mark_mine)
c.pack()
for i in range(GRID_SIZE):
    for j in range(GRID_SIZE):
```

c.create_rectangle(i * SQUARE_SIZE, j * SQUARE_SIZE, i * SQUARE_SIZE + SQUARE_SIZE,

```
                        j * SQUARE_SIZE + SQUARE_SIZE, fill='gray')
root.mainloop()
```

Нәтижесі:

- Тапсырма:
- Тор жасап, миналар қою
- Экранда 5×5 өлшеміндегі тор жаса (немесе қалауыңа қарай 6×6, 8×8).
- Әр ұяшыққа координаталарға сәйкес тікбұрыш сыз.
- Мина (💣) кездейсоқ ұяшықтарға жасырын түрде орналастырылсын.

- Тышқанмен бір ұяшықты басқанда:
- Егер ол жерде мина жоқ болса - ашық түс көрсетілсін;
- Егер мина болса - ойын аяқталсын. Ескерту: Turtle графикасында бұл ойынды толық жасау қиын, бірақ қарапайым логикалық модель жасау арқылы түсінік беріледі. Толық GUI ойыны үшін tkinter не pygame қолайлы.

Дескриптор:

- Tkinter кітапханасында программа құрады.
- Программаны талдайды.
- Тор жүйесін (квадрат ұяшықтар) дұрыс салады
- Миналарды кездейсоқ орналастырады
- Тышқанмен шерткенде логикалық жауап пайда болды
- Мина ашылғанда "Ойын аяқталды" деген хабарлама шықты

Сұрақтар:

```
•    random.randint() функциясы не үшін қолданылды?
```

- Ұяшыққа тышқанмен басқанда қандай координаталар есептеледі?
- Мина мен бос ұяшықты қалай ажыратамыз?
- Егер миналар бір-бірінің үстіне түссе, не болады? Қалай алдын аламыз?
- onclick() функциясы қандай оқиғаға жауап береді?
- Жеңу шартын қалай қосуға болады?
- Ұяшықтар ашылғанда көршілерінде қанша мина бар екенін қалай тексеруге болады?
- Экранда "ойын қайта бастау" батырмасын қалай қосар едіңіз?
- Көп деңгейлі (beginner, intermediate, expert) ойын жасау үшін не өзгерту керек?
- Қосымша дыбыс немесе анимация қосқыңыз келсе, қандай модуль қолданар едіңіз?

Практикалық жұмыс 9: GUI арқылы программаларды басқару

GUI кітапханасы және Label элементі. 1980-жылдардың соңына қарай коммерциялық операциялық жүйелерде жаңа типтегі интерфейс - графикалық қолданушы интерфейсі (ағылш. Graphical User Interface, қысқаша GUI) пайда болды. Бұл интерфейс пайдаланушы мен операциялық жүйе арасындағы байланысты экранда көрінетін графикалық элементтер арқылы жүзеге асыруға мүмкіндік берді. GUI жүйелері тышқан құрылғысын мәлімет енгізу үшін кеңінен қолдана бастады. Бұрынғыдай мәтіндік командалар терудің орнына, пайдаланушы белгілі бір нысанды жай ғана шертіп, әрекет орындай алатын болды. Қажетті функциялар мен параметрлер шағын сұхбаттық терезелер арқылы ыңғайлы түрде орындала бастады. Python бағдарламалау тілінде графикалық интерфейстермен жұмыс істеуге мүмкіндік беретін арнайы Tk модулі қолданылады. Бұл модульдің негізі саналатын Tk кітапханасы басқа да көптеген бағдарламалау тілдерінде кең таралған. Python-да GUI-мен жұмыс істеуге арналған бірқатар компоненттер ендірілген. Осы компоненттер арқылы әртүрлі графикалық интерфейсі бар терезелер құруға болады. Бұл терезелерде пайдаланушы Label, батырма (Button), енгізу өрісі (Entry) сынды интерфейс элементтері арқылы бағдарлама жұмысына тікелей ықпал ете алады. GUI орнату. Командалық жолдың интерфейсі. Label интерфейсінің элементі терезеге мәтін шығару үшін қолданылады. Ол бір жол мәтін немесе жазба (жазу) шығарады. • Label интерфейсінің элементін жасау үшін tkinter модулінің Label класы экземплярын құру керек.

№ 1 Мәтінмен жұмыс Label интерфейсі элементі бар терезе шығарады, ол экранда "Әлемге сәлем!" жазуын бейнелейді

```
# Программа 13.3 (hello_world.py)
```

Бұл программа мәтіні бар жазу көрсетеді.

```
import tkinter
class MyGUI:
def __init__(self) :
```

Басты терезе интерфейсінің элементін құру.

```
self.main_window = tkinter.Tk()
```

Label интерфейсінің 'Әлемге сәлем!' жазуы # бар элементін құру

```
 self.label = tkinter.Label(self.main_window,
text= 'әлемге сәлем!' )
```

Label интерфейсі элементінің pack әдісін шақыру.

```
self.label.pack()
```

tkinter ортасының басты цикліне кіру.

```
tkinter.mainloop()
```

MyGUI класы экземплярын жасау.

```
my_gui = MyGUI ()
```

Нәтижесі:

• • • •

- Тапсырмалар:
- 1 Екі санды қосу программасы
- Терезе (Tk()) ашыңыз.
- Қолданушы екі сан енгізетін Entry өрісін орнатыңыз.
- «Қосу» батырмасы болсын.
- Батырма басылғанда сандар қосылып, нәтижесі Label арқылы шықсын. •
- 2 Батырма арқылы терезе түсін өзгерту
- Терезеде бірнеше батырма болсын: "Қызыл", "Жасыл", "Көк".
- Қай батырма басылса, терезе сол түске боялсын.

Дескриптор

- Tkinter кітапханасында программа құрады.
- Tkinter кітапханасында мәтінмен жұмыс жасай алады.
- GUI терезесін жасайды (Tk() қолданады)
- Екі Entry өрісі арқылы дерек қабылдайды
- Батырмаға функцияны байланыстырады (command=...)
- Нәтижені экранда Label арқылы көрсетеді
- Қате енгізуге қарсы тексеруді қосады (try/except)
- Батырмаларға command дұрыс байланыстырады
- Терезе түсін батырмаға сәйкес өзгертеді

- lambda арқылы аргумент беру тәсілін қолданады

Сұрақтар:

```
    •    Интерфейс элементнің pack() әдісі не істейді?
```

- Егер Label интерфейсінің екі элементін жасап, олардың pack() әдістерін аргументсіз шақырсақ, интерфейстің негізгі (аталық) элементінде Label элементтері қалай орналасады ?
- Негізгі интерфейстің (аталық) элементінде орналастырылып жатқан интерфейс элементі барынша сол жақ шетке таман орналасуы үшін, pack() әдісіне қандай аргумент беру керек?
- tk.Entry() мен tk.Label() қандай айырмашылығы бар?
- command=add_numbers деген не істейді?
- Егер қолданушы сан орнына мәтін енгізсе не болады? Қалай алдын алдың?
- pack() әдісі қандай рөл атқарады?
- get() функциясы қандай типтегі мән қайтарады?
- Егер батырманы екі рет бассақ, не болады? Неліктен?
- Нәтижені Label орнына MessageBox арқылы шығару үшін не істеу керек?
- Санның орнына «өнімділік» немесе «жылдамдық» сияқты сөздік деректер енгізіп салыстыру мүмкін бе?
- GUI арқылы енгізілген деректерді .txt файлына жазу үшін не өзгерту қажет?
- Бірнеше батырма (қосу, азайту, көбейту, бөлу) қосқыңыз келсе, не өзгерту керек?

Практикалық жұмыс 10: Frame жақтаулары интерфейсі

Frame интерфейсінің элементі контейнер болып табылады, оның ішінде интерфейстің басқа элементтерін де сақтауға болады. Frame жақтаулары терезедегі интерфейс элементтерін реттейді және оларды топтап орналастыру үшін де қолданылады. Мысалы, интерфейс элементтері жиынын бір Frame жақтаулары ішінде белгілі бір тәртіппен орналастырып, сонан кейін оларды өзге Frame ішіне басқаша тәртіппен орналастыра аламыз. № 1 Екі түрлі жақтауларда жазу шығару программасының коды

```
# Программа 13.6 (frame_demo.py)
```

Бұл программа екі түрлі жақтауларда жазу жазады.

```
import tkinter
class MyGUI:
def __init__(self):   # Басты терезе интерфейсінің элементін құру.
```

self.main_window = tkinter.Tk() # Екі жақтау құру: біреуін терезенің жоғарғы бөлігі үшін # екіншісін төменгі бөлігі үшін.

```
self.top_frame = tkinter.Frame(self.main_window)
self.bottom_frame = tkinter.Frame(self.main_window)
```

Жоғарғы жақтау үшін Label интерфейсінің # үш элементін құру.

```
self.label1 = tkinter.Label(self.top_frame,
text='Ым жасау')
self. label2 = tkinter.Label(self.top_frame,
text ='Көз қысу')
self.label3 = tkinter.Label(self.top_frame,
text ='Бас изеу' )  # Жоғарғы жақтауда орналасқан жазуларды сатылау.
```

Оларды бірінен кейін бірін орналастыру үшін # side='top' аргументін қолдану

```
self.label1.pack(side='top')
self.label2.pack(side='top')
self.label3.pack(side='top')
```

Төменгі жақтау үшін Label интерфейсінің # үш элементін жасау.

```
self.label4 = tkinter.Label(self.bottom_frame, text ='Ым жасау')
self. label5 = tkinter.Label(self.bottom_frame,  text = 'Көз қысу' )
self.label6 = tkinter.Label(self.bottom_frame,  text ='Бас изеу')
```

Төменгі жақтаудағы жазуларды тығыздау. #Оларды көлденең сол жақта шетке орналастыру үшін # side='left' аргументін қолдану.

```
self.label4.pack(side='left')
self.label5.pack(side='left')
self.label6.pack(side='left') # Жақтауларды да тығыздап орналастыру!
self.top_frame.pack()
self.bottom_frame.pack() # tkinter ортасының басты цикліне кіру.
 tkinter.mainloop()
 my_gui = MyGUI() # MyGUI класының экземплярын жасау.
```

Нәтижесі: • • • • • • • • • • •

- Тапсырмалар:
- 1 Қарапайым интерфейсті жақтауларға бөлу
- Терезе (Tk) ашыңыз.
- Интерфейсті 2 бөлікке бөліңіз: жоғарғы жақтауда (Frame1) - тақырып пен мәтін, төменгі жақтауда (Frame2) - батырма болсын.
- Батырма басылған кезде мәтін жаңарсын. •
- 2 Көпжақтаулы интерфейс
- Терезені 3 бөлікке бөліңіз (top_frame, middle_frame, bottom_frame);
- top_frame - қолданушы аты енгізетін өріс;
- middle_frame - жынысын таңдайтын радиобатырмалар;
- bottom_frame - «Қабылдау» батырмасы, басылғанда нәтижені көрсетсін.
- Қалалар тізімі бар Frame жасаңыз және таңдалған қала бойынша ақпарат шығарыңыз.
- Әр жақтауда түрлі түстер, өлшемдер және виджеттер арқылы интерфейсті сәндеңіз.
- Тіл ауыстырғыш немесе тақырып өзгертуші батырмаларды жақтаулар ішінде ұйымдастырыңыз.

Дескриптор

- Frame жақтауларын пайдалана алады.
- Tkinter кітапханасында мәтінмен жұмыс жасай алады.
- Әр Frame ішінде элементтерді орналастырады
- Батырма басылғанда Label мәтінін жаңартады
- Интерфейсті эстетикалық түрде орналастырады Сұрақтар:
- Frame виджеті не үшін қолданылады?
- Бір терезеде бірнеше Frame қолдануға бола ма?
- fill="both" және expand=True параметрлері не істейді?
- Frame ішінде орналасқан Label мен Button қалай байланысады?
- Бір Frame ішінде бірнеше жолдық элементтерді орналастыру үшін қандай әдіс қолданасыз?
- Егер батырма басқа Frame ішінде болса, сырттағы элементке қалай әсер ете алады?

- pack() әдісімен қатар grid() немесе place() қолдануға бола ма?
- Frame өлшемін нақты (пиксельмен) көрсету үшін не істеу керек?
- Бір интерфейсте 3 немесе одан көп Frame қолдансаңыз, қандай қиындық туындауы мүмкін?
- Frame ішінде орналасқан элементтерді топтап жасыру немесе көрсету мүмкін бе?

Практикалық жұмыс 11: Button интерфейсі

Button интерфейсі элементтері терезеде стандартты батырма жасау үшін қолданылады. Қолданушы батырманы басқан кезде берілген функция немесе әдіс шақырылады.

- Ақпараттық сұқбаттасу терезесі - бұл қарапайым терезе, ол мәлімет шығарады және шерткен кезде осы терезені жабатын OK батырмасы бар. Ақпараттық сұқбаттасу терезесі шығару үшін tkinter.messagebox модулінің showinfo функциясы қолданылады.
- Button - бұл шерткен кезде белгілі бір әрекет орындауды жүзеге асыратын интерфейс элементі. Бұл Button интерфейсі элементінің, яғни батырманың бетіне мәтін жазып, оған қоса кері шақыру функциясын да көрсете аламыз. Кері шақыру функциясы - бұл батырманы шерткен сәтте орындалатын функция немесе әдіс.
- Кері шақыру функциясын оқиға өңдеуіші (event handler) деп те атайды, өйткені ол батырманы шерткен сәтте орындалатын оқиғаны өңдейді.
- Ақпараттық сұқбат терезесін шығару үшін tkinter.messagebox модуліндегі showinfo функциясы қолданылады. Ол функцияны пайдалану үшін, tkinter.messagebox модулін импорттау қажет. Showinfo функциясын шақыру жолы: tkinter.messagebox.showinfo(тақырып, хабарлама) Бұл форматтағы тақырып - бұл экранға тақырып маңында шығарылатын сөз тіркесі, хабарлама - бұл да тіркес, сұқбат терезесінің негізгі бөлігенде бейнелетін мәтін.

№1 Button интерфейсі пайдаланып мәтін шығару. # Бұл программа элементін көрсетеді. Қолданушы Button батырмасын шерткенде, экранға ақпараттық сұқбат терезесі шығады.

```
import tkinter
import tkinter.messagebox
class MyGUI:
def __init__(self):  # Басты терезе интерфейсінің элементі.
self.main_window = tkinter.Tk() # Button widget интерфейсінің элементін құру.
```

#Қолданушы батырманы шерткенде, # батырмада 'Мені шерт! мәтіні пайда болады да, # do_something әдісі орындалуы тиіс.

```
self.my_button = tkinter.Button(self.main_window,
text='Мені шарт!',
command=self.do_something)    # Button интерфейсі элементін тығыздау
 self.my_button.pack() # Басты tkinter цикліне кіру.
tkinter.mainloop()       # do_something әдісі Button интерфейсі элементі үшін
def do_something(self):       # Ақпараттық сұқбат терезесін көрсету.   tkinter.messagebox.showinfo('Реакция',
'Батырманы шерткеніңіз үшін, алғыс.' )
```

MyGUI класының экземплярын құру.

```
my_gui = MyGUI()
```

Нәтижесі:

• • • • • • •

- Тапсырмалар:
- 1 Батырма басқанда мәтін шығару
- Терезе (Tk()) ашыңыз.
- Батырма орнатыңыз (мысалы, "Бас" деп жазу).
- Батырма басылғанда экранда "Сәлем, әлем!" деген жазу Label арқылы шықсын. •

- 2 Қосу батырмасы
- Екі Entry өрісіне сан енгізіңіз.
- «Қосу» батырмасын басқанда, нәтижесі экранда көрсетілсін.

Дескриптор

- Button пайдалана алады.
- Tkinter кітапханасында мәтінмен жұмыс жасай алады.
- Терезе (Tk) құрады
- Батырманы (Button) интерфейске орнатады
- Батырмаға функцияны (command) байланыстырады
- Батырма басылғанда нәтижені экрғада шығарады (Label)
- Екі Entry өрісін құрады
- Батырма арқылы енгізілген сандарды қосады
- Нәтижені экранда Label арқылы шығарады
- Қате енгізуді өңдейді (try/except)

Сұрақтар:

```
    •    tk.Button() қандай мақсатта қолданылады?
```

- command=say_hello деген нені білдіреді?
- label.config(text=...) деген не үшін қажет?
- pack() әдісін қолданбаса, батырма көріне ме?
- Бір батырма бірнеше әрекет орындауы мүмкін бе?
- Егер command ішінде жақша қойсақ (мысалы, command=say_hello()), не болады?
- Button дизайнын (түс, қаріп, өлшем) қалай өзгертуге болады?
- Бір терезеде бірнеше батырма қолдансаңыз, оларды қалай реттеп орналастырасыз?
- Батырма басылған сайын санды 1-ге арттыру үшін не істеу керек?
- Button арқылы басқа терезе ашуға бола ма?

Практикалық жұмыс 12: «Шығу» батырмасын салу

GUI бар программада терезені жабатын Шығу (немесе Cancel - Отмена батырмасы) болады. Шығу батырмасын құру үшін, Button интерфейсі элементі жасалады, ол кері шақыру функциясы ретінде интерфейстің негізгі элементінің destroy() әдісін шақырады.

№1 Frame жақтауларын пайдаланып «Шығу» батырмасын салу программасының коды # Бұл программада 'Шығу' батырмасы бар, # оны шерткенде, ол Tk класының destroy әдісін шақырады.

```
import tkinter
import tkinter.messagebox
class MyGUI:
def __init__(self) :
```

Басты терезенің интерфейсі элементін жасау.

```
self.main_window = tkinter.Tk()
```

Button widget интерфейсі # Батырмада 'Мені шерт!' мәтіні шығуы тиіс. # Қолданушы батырманы шерткенде, # do_something әдісі орындалуы тиіс.

```
self.my_button = tkinter.Button(self.main_window,
text = 'Мені шерт!',
command = self.do_something)
```

'Шығу' батырмасын жасау. Бұл батырманы шерткенде, # интерфейстің түпкі элементінің destroy әдісі шақырылады # (main_window айнымалысы түпкі элементке сілтеме жасайды, # сондықтан кері шақыру функциясы self.main_window.destroy болады.)

```
 self.quit_button = tkinter.Button(self.main_window,
text = 'Шығу',
```

command = self.main_window.destroy # Button интерфейсі элементін тығыздау.

```
self.my_button.pack()
self.quit_button.pack()      # Басты tkinter цикліне кіру.
tkinter.mainloop()    # Button интерфейсінің элементі үшін, do_something әдісі
```

кері шақыру функциясы болып табылады .

```
 def do_something(self):       #Ақпараттық сұқбат терезесін көрсету.
 tkinter.messagebox.showinfo('Реакция',  'Батырманы шерткеніңізге алғыс. ')
 my_gui = MyGUI() # MyGUI класы экземплярын құру.
```

Нәтижесі:

• • •

- Тапсырма:
- 1 Шығу батырмасын жасау
- Интерфейс терезесін (Tk()) ашыңыз;
- Терезеде бір ғана батырма болсын - «Шығу»;
- Батырма басылғанда бағдарлама жабылсын. •
- 2 Шығу алдында растау сұрау
- Шығу батырмасы басылғанда, қолданушыдан растау сұралсын.
- Егер «Иә» десе - бағдарлама жабылады, «Жоқ» десе - жұмыс жалғасады.

Дескриптор:

- Button,Frame жақтауларын пайдалана алады.
- Tkinter кітапханасында мәтінмен жұмыс жасай алады.
- Терезе (Tk) құрады
- Батырманы интерфейске дұрыс орналастырады
- Батырмаға command арқылы функция байланыстырады
- Батырма басылғанда терезені жабады (destroy())
- Батырманың түсін және мәтінін көрнекі түрде безендіреді

Сұрақтар:

```
    •    destroy() мен quit() функцияларының айырмашылығы қандай?
```

- command=exit_app не істейді?
- Егер батырмада command болмаса не болады?

- fg, bg, font параметрлері не үшін қажет?
- Батырма терезенің ортасында орналасу үшін қандай әдіс қолданылды?
- Терезені жабу үшін батырмадан басқа қандай тәсілдер бар?
- Егер бірнеше батырма болса, әрқайсысына әртүрлі әрекет тағайындауға бола ма?
- «Шығу» батырмасын басқанда растау сұрау (confirmation) үшін не істеу керек?
- messagebox.askyesno() арқылы шығуды растау мүмкін бе?
- Батырманы тек белгілі бір шартпен ғана белсенді етуге бола ма?

- «Иә» жауап бергенде бағдарлама жабылды
- «Жоқ» десек - бағдарлама жалғасты

Практикалық жұмыс 13: Entry интерфейсі

Entry интерфейсінің элементі - бұл GUI программасына мәлімет енгізуге болатын тіктөртбұрышты аймақ. Entry элементіне енгізілген мәнді алу үшін get() әдісі қолданылады.

- Көбінесе, программада бір немесе бірнеше батырмасы бар Entry элементтері болады, оларды қолданушы енгізілген мәндерді алу үшін шертеді. Батырманың кері шақыру функциясы терезедегі Entry элементінен мәндерді алып, соларды өңдейді.
- Entry интерфейсі элементінен мәндерді алу үшін оның get() әдісі қолданылады. Бұл әдіс тіркестік мән қайтарады, сондықтан оны керекті типке келтіріу қажет. Мысалы, Программада Entry интерфейсі элементіне енгізілген мән нақты санға түрлендіріледі. Бұл интерфейс элементтерін көрсету үшін, қолданушы программадағы Entry элементіне километрмен берілген аралықты енгізеді, сонан соң батырманы шертеді, сонда сол аралық мильмен көрсетіледі. Километрді мильге түрлендіру формуласы: мильдер = километрлер x 0.6214. Интерфейс элементтерін бекітілген орындарға қою үшін біз оларды екі жақтауға орналастырамыз. Көмекші мәтін тұратын Label элементі мен Entry элементі жоғарғы top_frame жақтауында болады, олардың pack() әдістері side='left' аргументі арқылы шақырылады. Осының нәтижесінде олар жақтауда көлденең орналасады. Түрлендіру және Шығу батырмалары төменгі bottom_frame жақтауында болып, олардың pack() әдістері де side='left' аргументі арқылы шақырылады.

№1 Entry элементіне 1000 санын енгізіп, оны мильге түрлендіру. #Бұл программа километрмен берілген аралықты мильге түрлендіреді. Алынған нәтиже ақпараттық терезеге, яғни сұқбаттасу терезесіне шығарылады.

```
import tkinter
import tkinter.messagebox
class KiloConverterGUI:
def __init__(self) :       # Бас терезе құру. ін
self.main_window = tkinter.Tk()  # Интерфейс элемент топтау үшін, екі жақтау жасау.
self.top_frame = tkinter.Frame(self.main_window)
self.bottom frame = tkinter.Frame(self.main_window)
```

Жоғарғы жақтау үшін интерфейс элементтерін құру.19 self.prompt_label = tkinter. Label(self.top_frame,text='Аралықты километрмен енгізіңіз: ')

```
self. kilo_entry = tkinter.Entry(self.top_frame, 22 width=10)
```

Жоғарғы жақтау элементтерін нықтау.

```
self.prompt_label.pack(side='left')
self.kilo_entry.pack(side='left') # Button интерфейсі элементін құру.
self.calc_button = tkinter.Button(self.bottom_frame,
text='Түрлендіру',
command=self.convert)
self.quit_button = tkinter.Button(self.bottom_frame,
text='Шығу',
command=self.main_window.destroy) # Батырмаларды көлденең бағытта нықтау.
self.calc_button.pack(side='left')
self.quit_button.pack(side='left') # Жақтауларды да көлденең бағытта нықтау.
```

```
self.top_frame.pack()
self.bottom_frame.pack()  # Басты tkinter цикліне кіру.
tkinter.mainloop()      # convert әдісі 'Түрлендіру' батырмасы үшін
```

кері шақыру функциясы болып табылады.

```
def convert(self):         # kilo_entry интерфейсі элементіне
 kilo = float(self.kilo_entry.get())
```

Километрлерді мильдерге түрлендіру. miles = kilo * 0.6214 # Нәтижелерді ақпараттық сұқбат терезесінде көрсету. tkinter.messagebox.showinfo('Нәтижелері', str(kilo) + ' километр ' + str (miles) + ' мильге тең.') # KiloConverterGUI класы экземплярын құру.

```
 kilo_conv = KiloConverterGUI()
```

Нәтижесі:

Сұхбат терезе •

- Тапсырма:
- 1 Атыңды енгіз және сәлемде
- Қолданушы Entry өрісіне өзінің атын енгізеді.
- «Сәлемдесу» батырмасын басқанда, экранда "Сәлем, [аты]!" деген жазу шығады. •
- 2 Құпия сөз тексеру
- Қолданушы Entry өрісіне құпия сөз енгізеді.
- «Тексеру» батырмасын басқанда:
- Егер құпия сөз дұрыс болса - "Қош келдіңіз!"
- Қате болса - "Құпия сөз қате!" деген жазу шығару.

Дескриптор:

- Button,Frame жақтауларын пайдалана алады.
- Tkinter кітапханасында мәтінмен жұмыс жасай алады.
- Entry өрісін интерфейске дұрыс енгізеді
- Батырманы Entry-мен байланыстырады
- Қолданушы енгізген деректі Label-де көрсетеді
- Интерфейсті логикалық түрде құрады
- Entry өрісінде show="*" параметрін қолданады
- Батырма арқылы құпия сөзді тексереді
- Нәтижені экранда Label арқылы көрсетеді
- Қате енгізуге қарсы жауап береді

Сұрақтар:

- Entry мен Label айырмашылығы қандай?
- .get() әдісі не үшін қолданылады?
- Қолданушы ештеңе енгізбесе, не болады?
- Entry-ге бастапқы мән (default value) беруге бола ма?
- Құпия сөз енгізу үшін Entry қалай өзгертіледі?
- Бірнеше Entry өрісімен қалай жұмыс істеуге болады?
- Енгізілген деректі тек сандар немесе тек әріптер ретінде тексеруге бола ма?
- .delete() және .insert() әдістері не үшін керек?

- Егер қолданушы тек бос орын енгізсе, не істеу керек?
- Entry өрісіне шектеу (максимум 20 символ, тек цифрлар) қоюға бола ма?

Практикалық жұмыс 14: Label элементі

StringVar объектісі Label интерфейсі элементімен байланысты болған жағдайа, Label интерфейсі экранға StringVar объектісінде сақталатын кез келген мәліметті шығара алады. • Мәліметтерді шығару үшін, ақпараттық сұқбат терезесін қолдандық. Басты терезеде Label бос элементтері жасалады да, соған арналған программалық код жазылып, батырманы шерткен кезде, керекті мәліметтер экранға шығарылады. Tkinter модулінде StringVar класы бар, ол мәліметтер шығару үшін Label интерфейсі элементімен бірге қолданылады. • Мұнда алдымен StringVar объектісін жасап алу қажет. Сонан соң Label интерфейсі элементін құрып, оны StringVar объектісімен байланыстырады. StringVar объектісінде сақталатын кез келген мән автоматты түрде Label элементіне шығарылады.

№1 Программа (kilo_converter2.py).т Бұл программа километрмен берілген аралықты мильге түрлендіреді. Алынған нәтиже бас терезедегі Label элементіне шығарылады.

```
import tkinter
 class KiloConverterGUI:
def __init__(self) : 9 10 # Бас терезе құру.
self.main_window = tkinter.Tk()  # Интерфейс элементін топтау үшін үш жақтау жасау.
self.top_frame = tkinter.Frame()
self.mid_frame = tkinter.Frame()
self.bottom_frame = tkinter.Frame()     # Жоғарғы жақтау  интерфейс элементін құру.
self.prompt_label = tkinter.Label(self.top_frame,
 text='Аралықты километрмен енгізіңіз:')
 self.kilo_entry = tkinter.Entry(self.top_frame, width=10)
```

Жоғарғы жақтау элементтерін нықтау.

```
 self.prompt_label.pack(side='left')
```

self.kilo_entry.pack(side='left') #Орталық жақтау интерфейс элементін құру.

```
 self.descr_label = tkinter.Label(self.mid_frame,
 text='Мильге түрлендірілді: ')
 self.value = tkinter.StringVar()               #Label жасап, оны StringVar байланыстыру
self.miles_label = tkinter.Label(self.mid_frame,
textvariable=self.value)                # Орталық жақтау үшін интерфейс элементін құру.
self.descr_label.pack(side='left') 46 self.miles_label.pack(side='left')
```

self.calc_button=tkinter.Button(self.bottom_frame,text='Түрлендіру', command=self.convert)

```
self.quit_button = tkinter.Button(self.bottom_frame,
text='Шығу' , command=self.main_window.destroy)
self.calc_button.pack(side='left')
self.quit_button.pack(side='left')         # Жақтауларды нықтау.
 self.top_frame.pack()
self.mid_frame.pack()
self.bottom_frame.pack()                   # Басты tkinter цикліне кіру.
```

tkinter.mainloop() # convert әдісі 'Түрлендіру' батырмасы үшін

```
def convert(self):                              # Қолданушының kilo_entry интерфейсі
kilo = float(self.kilo_entry.get())      # Километрлерді мильдерге түрлендіру.
miles = kilo * 0.6214
```

self.value.set(miles) # KiloConverterGUI класы экземплярын жасау.

```
kilo_conv = KiloConverterGUI()
```

Нәтижесі:

Программа нәтижелерінің Label элементтері арқылы экранға шығарылу нәтижелері

- Тапсырмалар:
- 1 Қарапайым мәтін шығару
- Интерфейс терезесін (Tk()) жасаңыз;
- Терезеде бір Label болсын - "Қош келдіңіз!" деген жазу шығарып тұрсын;
- Label қаріпі мен түсін өз қалауыңызша өзгертіңіз. •
- 2 Мәтінді батырма арқылы жаңарту
- Интерфейс терезесінде бір Label және бір Button болсын.
- Батырма басылғанда Label ішіндегі мәтін "Сәлем, Python!" деп жаңарсын.

---

3 БІРНЕШЕ ТҮЙМЕ ОРНАТЫП, ӘРҚАЙСЫСЫН БАСҚАНДА LABEL ТҮСІ ӨЗГЕРЕТІН БОЛСЫН.

Дескриптор:

- Button,Frame жақтауларын пайдалана алады.
- Tkinter кітапханасында мәтінмен жұмыс жасай алады.
- Терезе құрады (Tk())
- Label интерфейске дұрыс орнатады
- Label ішінде мәтінді шығарады
- Мәтіннің түсі немесе қаріпін өзгертеді
- Label және Button орналастырады
- Button басылғанда Label мәтіні өзгереді
- command арқылы дұрыс байланыс орнатады

Сұрақтар:

- Төменде келтірілген tkinter модулі интерфейсінің элементтерін қысқаша сипаттаңыз: а) Label; б) Entry; в) Button; г) Frame.
- Интерфейстің негізгі элементі қалай құрылады?
- tkinter модулінің mainloop функциясы не істейді?
- Label не үшін қолданылады?
- Label мен Entry айырмашылығы қандай?
- fg, bg, font параметрлері не үшін керек?
- Label элементін терезенің ортасына қалай орналастыруға болады?
- Бір терезеде бірнеше Label қолдануға бола ма?
- Label ішіндегі мәтінді өзгерту үшін қандай әдіс қолданылады?
- Қолданушы әрекетіне байланысты Label мәтіні қалай өзгереді?
- wraplength және justify параметрлерінің қызметі қандай?
- Егер терезе кішірейсе, Label қалай әрекет етеді?
- Label ішінде сурет көрсету мүмкін бе?

Практикалық жұмыс 15: GUI көмегімен программа құру

№1 Бағаның орташа мәнін шығару программасының коды. Оқушылардың үш пәннен алған бағаларының орташа мәнін шығарайық. Программаға барлық бағаларын енгізіп, солардың орташа мәнін батырма басылған кезде экранда көрсету. Сызбада интерфейсі

```
import tkinter
class TestAvg:
def __init__(self):                                # Басты терезе құру.
self.main_window = tkinter.Tk()           # Бес жақтау құру.
self.test1_frame = tkinter.Frame(self.main_window)
self.test2_frame = tkinter.Frame(self.main_window)
self.test3_frame = tkinter.Frame(self.main_window)
self.avg_frame = tkinter.Frame(self.main window)
self.button_frame = tkinter.Frame(self.main_window)
```

1-баға үшін интерфейс элементтерін құрып нықтау.

```
self.test1_label = tkinter.Label(self.test1_frame,text='1-бағаны енгізіңіз: ')
self.test1_entry = tkinter.Entry(self.test1_frame, width=10)
self.test1_label.pack(side='left')
```

elf.test1_entry.pack(side='left') # 2-баға үшін интерфейс элементтерін құрып нықтау. self.test2_label = tkinter.Label(self.test2_frame, text='2-бағаны енгізіңіз: ')

```
self.test2_entry = tkinter.Entry(self.test2_frame, 30 width=10)
self.test2_label.pack(side='left')
self.test2_entry.pack(side='left')           # 3-баға үшін интерфейс элементтерін құру.
self.test3_label = tkinter.Label(self.test3_frame, text='3-бағаны енгізіңіз:')
self.test3_entry = tkinter.Entry(self.test3_frame, width=10)
self.test3 label.pack(side='left') 40 self.test3_entry.pack(side='left')
```

Орташа балл үшін интерфейс элементтерін құрып нықтау.

```
self.result_label = tkinter.Label(self.avg_frame,text='Орташа балл:')
self.avg = tkinter.StringVar() # avg_label-ді жаңарту үшін
self.avg_label = tkinter.Label(self.avg_frame, textvariable=self.avg)
self.result_label.pack(side='left')
self.avg_label.pack(side='left')             # Button интерфейсі элементін құрып нықтау.
self.calc_button = tkinter.Button(self.button_frame,
text='Орталау', 54 command=self.calc_avg)
self.quit_button = tkinter.Button(self.button_frame,
text='Шығу' , 57 command=self.main_window.destroy)
self.calc_button.pack(side='left')
self.quit_button.pack(side='left')                              # Жақтауларды нықтау.
self.test1_frame.pack()
self.test2_frame.pack()
self.test3_frame.pack()
self.avg_frame.pack()
self.button_frame.pack()                                  # Басты циклді іске қосу.
```

tkinter.mainloop() # calc_avg әдісі calc_button интерфейсі үшін

```
def calc_avg(self):                  # Үш бағаны алып, оларды айнымалыларда сақтау.
self.test1 = float(self.test1_entry.get())
self.test2 = float(self.test2_entry.get())
self.test3 = float(self.test3_entry.get())                      # Орташа баллды есептеу.
self.average = (self.test1 + self.test2 + 83 self.test3) / 3.0
```

self.average мәндерін avg сілтеме жасап тұрған # StringVar объектісінде сақтап алып, # avg_label интерфейсі элементін жаңалау.

```
self.avg.set(self.average)                    # TestAvg класы экземплярын жасау
test_avg = TestAvg()
```

Нәтижесі:

• • •

- Тапсырмалар:
- 1 Қолданушыдан аты-жөнін сұрау
- Терезеде:
- Екі Entry: Аты және Тегі үшін;
- Бір Button: «Көрсету»;
- Бір Label: нәтижені шығару үшін;
- Қолданушы аты мен тегін енгізіп, батырма басқанда «Сәлем, [аты] [тегі]!» деп жазылуы керек. •
- 2 Қарапайым калькулятор (қосу ғана)
- Екі санды енгізу үшін Entry өрісі;
- «Қосу» батырмасы;
- Нәтиже Label арқылы көрсетілуі керек.

Дескриптор:

- Button,Frame жақтауларын пайдалана алады.
- Tkinter кітапханасында мәтінмен жұмыс жасай алады.
- Интерфейске кемінде екі Entry өрісті орналастырады
- Батырманы (Button) дұрыс құрады
- Қолданушы енгізген мәлімет Label арқылы көрсетеді
- GUI визуалды түрде дұрыс және түсінікті орналастырады
- Екі сандық Entry енгізеді
- Батырма басылғанда сандарды қосады
- Нәтижені Label арқылы көрсетеді
- Қате енгізуді өңдейді (try-except)

Сұрақтар:

- tkinter қандай кітапхана? Ол не үшін қолданылады?
- GUI мен CLI (командалық интерфейс) айырмашылығы қандай?
- Entry мен Label элементтері не үшін қажет?
- Батырма басылғанда функция қалай шақырылады?
- Қолданушы қате енгізсе (бос қалдырса), қалай өңдеуге болады?
- GUI терезесінің өлшемін, түсін қалай өзгертуге болады?
- Бірнеше терезе (мысалы, екінші бет) ашуға бола ма?
- place(), pack(), grid() әдістерінің айырмашылығы қандай?
- Қолданушы енгізген деректі файлға сақтауға бола ма?
- Бір GUI ішінде бірнеше виджеттер қалай өзара әрекеттеседі?

Тест сұрақтары

- Python дегеніміз не?
  A) Операциялық жүйе
  B) Бағдарламалау тілі

C) Браузер
D) Мәтіндік редактор

- Python-да айнымалы қалай жарияланады?
A) var x = 5
B) let x = 5
C) x := 5
D) x = 5
- Python интерпретаторы қалай іске қосылады?
A) python
B) run python
C) py.start()
D) launch python
- Python-да ескертпе (комментарий) қандай таңбамен басталады?
A) //
B) <!--
C) #
D) /**
- type(3.14) нәтижесі қандай типті қайтарады?
A) int
B) str
C) float
D) bool
- print("Hello" + "World") нәтижесі:
A) Hello + World
B) HelloWorld
C) Hello World
D) HelloWorld()
- Қайсысы дұрыс айнымалы атауы?
A) 2value
B) my-var
C) _value1
D) class
- Айнымалы мәнін жою үшін қандай команда қолданылады?
A) erase
B) del
C) delete
D) remove
- input() функциясы не қайтарады?
A) int
B) float
C) bool
D) str
- int("5") + 3 нәтижесі қандай болады?
A) "53"
B) 8
C) 5.3
D) Қате
- Арифметикалық оператор қайсы?
A) =
B) ==
C) +
D) and
- 5 % 2 нәтижесі қандай?
A) 0

B) 1
C) 2
D) 2.5

- Қайсысы салыстыру операторы?
A) =
B) :=
C) ==
D) ++
- True and False нәтижесі:
A) True
B) False
C) None
D) 0
- not True мәні қандай?
A) True
B) False
C) None
D) 1
- x = 3; x += 2 соңында x-тің мәні қандай?
A) 3
B) 5
C) 6
D) 2
- 10 // 3 операциясы не қайтарады?
A) 3.33
B) 3
C) 4
D) 1
- x = 4; print(x == 4) не басып шығарады?
A) 4
B) True
C) False
D) x
- 5 != 3 нәтижесі қандай?
A) True
B) False
C) Error
D) None
- x = 1; y = 2; x > y or x < y нәтижесі:
A) True
B) False
C) None
D) Error •
- Жауаптар тізімі (1-20): • • • • • • • • • • • • • B D A C C B C B D B

• • • • • • • • • •

C B C B B B B B A A

```
    •    len("Python") не қайтарады?
A) 5
B) 6
C) 7
D) 0
    •    "python".upper() нәтижесі:
A) PYTHON
```

```
B) Python
C) python
D) error
    •    "HELLO".lower() нәтижесі:
A) hello
B) HELLO
C) Hello
D) hELLO
    •    "Hello"[1] символы қандай?
A) H
B) e
C) l
D) o
    •    "abcde"[2:4] срезі не қайтарады?
A) ab
B) cd
C) bc
D) c
```

- ["a", "b", "c"].append("d") нәтижесінде не болады?
  A) ['a', 'b', 'c']
  B) ['a', 'b', 'c', 'd']
  C) ['d', 'a', 'b', 'c']
  D) ['a', 'b', 'c'].append('d')
- my_list = [1, 2, 3]; my_list[0] = 5 соңғы нәтиже қандай?
  A) [1, 2, 3]
  B) [5, 2, 3]
  C) [1, 5, 3]
  D) [0, 2, 3]
- list1 = [1, 2]; list2 = [3, 4]; list1 + list2 нәтижесі:
  A) [1, 2, 3, 4]
  B) [4, 3, 2, 1]
  C) [1, 2][3, 4]
  D) Error
- list("abc") не қайтарады?
  A) ['abc']
  B) ['a', 'b', 'c']
  C) ['a b c']
  D) 'a', 'b', 'c'
- my_list = [1, 2, 3]; my_list.pop() не қайтарады?
  A) 1
  B) 2
  C) 3
  D) Error
- for i in range(3): print(i) қандай сандарды басып шығарады?
  A) 1 2 3
  B) 0 1 2
  C) 0 1 2 3
  D) 1 2
- while циклі қай кезде тоқтайды?
  A) мәңгі жалғасады
  B) шарт жалған болғанда
  C) шарт шын болғанда
  D) break қолданғанда ғана
- break кілтсөзі не істейді?
  A) циклді жалғастырады
  B) циклді тоқтатады

C) бір айналым өткізеді
D) кодты тазалайды

- continue кілтсөзі не істейді?
A) циклді тоқтатады
B) циклді қайта бастайды
C) ағымдағы итерацияны өткізіп жібереді
D) мәнді арттырады
- if x > 5: блогы қандай жағдайда орындалады?
A) x ≤ 5
B) x = 5
C) x > 5
D) x == 0
- if...elif...else қандай құрылым?
A) цикл
B) шарт
C) функция
D) модуль
- range(1, 5) не қайтарады?
A) 1 2 3 4 5
B) 1 2 3 4
C) 1 2 3
D) 0 1 2 3
- for циклі қай құрылымдарда қолданылады?
A) тек тізімдер
B) тек жолдар
C) тек сандар
D) тізім, жол, кортеж және т.б.
- if not x: не білдіреді?
A) x == True
B) x жоқ немесе False
C) x == 0
D) x міндетті түрде бар
- x = 3; y = 4; if x < y: қандай нәтиже береді?
A) True
B) False
C) Қате
D) y < x •
- Жауаптар тізімі (21-40): • • • • • • • • • • • • • •

B A A B D B B A B C

• • • • • • • • • •

B B B C C B B D B A

• •

```
    •    Функция қалай анықталады?
A) function myFunc():
B) def myFunc():
C) define myFunc():
D) func myFunc():
    •    return операторының мақсаты қандай?
A) экранға шығарады
B) циклді тоқтатады
```

```
C) мән қайтарады
D) қате жібереді
```

- Функция аргументі деген не?
  A) өзгермелі мән
  B) функция атауы
  C) сыртқы айнымалы
  D) функцияға берілетін мән
- Әдепкі аргументі бар функция қалай жазылады?
  A) def func(x: 5):
  B) def func(x = 5):
  C) def func(5 = x):
  D) def func x = 5:
- *args не үшін қолданылады?
  A) функцияны қайталау
  B) шексіз аргументтер қабылдау
  C) тек бір мән беру
  D) қайтарымсыз функция жасау
- Аноним функция деген не?
  A) атауы жоқ функция
  B) клон функция
  C) айнымалы
  D) модуль
- Lambda функциясының синтаксисі қандай?
  A) def x: return x+1
  B) lambda x: x+1
  C) func x ⇒ x+1
  D) x ⇒ lambda(x+1)
- map() функциясы не істейді?
  A) сөздіктерді қосады
  B) циклді тоқтатады
  C) әр элементке функция қолданады
  D) мәнді бөледі
- filter() функциясы:
  A) барлық мәнді алып тастайды
  B) шартқа сәйкес элементтерді қалдырады
  C) тізімді сұрыптайды
  D) None қайтарады
- Функция ішіндегі global сөзі не үшін керек?
  A) айнымалыны жояды
  B) айнымалыны жасайды
  C) айнымалыны локализациялайды
  D) сыртқы айнымалыны өзгертеді
- Tuple дегеніміз не?
  A) Өзгермелі тізім
  B) Өзгермейтін тізім
  C) Мәтін
  D) Сөздік
- (1, 2, 3)[1] не қайтарады?
  A) 1
  B) 2
  C) (1, 2)
  D) 3

- Set коллекциясының негізгі ерекшелігі:
  A) Индекстеледі
  B) Қайталанатын мәндер болады
  C) Қайталанатын мәндер жоқ
  D) Мәтіндермен жұмыс істейді
- set1 = {1, 2, 3}; set1.add(4) нәтижесі қандай?
  A) {1, 2, 3}
  B) {4, 1, 2, 3}
  C) {1, 2, 3, 4}
  D) Error
- Қайсысы set әдісі?
  A) pop()
  B) append()
  C) insert()
  D) update()
- Dictionary кілттері қандай тип болуы керек?
  A) тек str
  B) тек int
  C) өзгермейтін типтер
  D) кез келген тип
- dict = {"a":1, "b":2}; dict["a"] нәтижесі:
  A) "a"
  B) 1
  C) "1"
  D) Error
- dict.get("x") егер "x" кілті жоқ болса:
  A) None
  B) 0
  C) Error
  D) "x"
- Сөздік элементін қалай өшіреміз?
  A) dict.remove("a")
  B) delete dict["a"]
  C) del dict["a"]
  D) erase dict("a")
- dict.keys() не қайтарады?
  A) Тек мәндерді
  B) Кілттердің тізімін
  C) Кілт-мән жұбын
  D) Сөздік ішіндегі барлық элементті

```
    •   Жауаптар тізімі (41–60):
```

• • • • • • • • • • • •

B C D B B A B C B D

• • • • • • • • • • •

B B C C D C B A C B

• •

- Python-да файл ашу үшін қандай функция қолданылады?
  A) open()
  B) fopen()

C) file()
D) readfile()

- Файлды тек оқу режимінде ашу үшін қандай аргумент жазылады?
A) "o"
B) "read"
C) "r"
D) "w"
- Файлды жазу режимінде ашу:
A) "r"
B) "w"
C) "rw"
D) "a"
- with open("data.txt", "r") as f: не үшін қолданылады?
A) Цикл жасау үшін
B) Файлды автоматты түрде жабу үшін
C) Модуль қосу үшін
D) Айнымалы жасау үшін
- f.read() не қайтарады?
A) Бір жол
B) Бір символ
C) Файлдың толық мәтіні
D) Сөздік
- f.readline() қандай мән қайтарады?
A) Соңғы жол
B) Алғашқы жол
C) Барлық жол
D) Бір символ
- f.readlines() не істейді?
A) Файлды тізім ретінде жолдарға бөледі
B) Барлық файлды өшіреді
C) Мәтінді блокпен оқиды
D) Файлды сұрыптайды
- Файлға жазу үшін қандай әдіс қолданылады?
A) write()
B) append()
C) insert()
D) save()
- "a" режимінде файл ашу не істейді?
A) Оқу
B) Жазуды өшіреді
C) Соңына қосады
D) Тек оқуға рұқсат береді
- Файл жолын көрсеткенде қандай слэш таңбасы қолданылады?
A) \
B) /
C) //
D) :
- try/except блогы не үшін қажет?
A) Кодты орнату үшін
B) Қате өңдеу үшін
C) Шарт қою үшін
D) Функция жасау үшін
- try: блогы дұрыс орындалмаса не орындалады?
A) except

B) break
C) if
D) assert

- Егер except блогы болмаса, не болады?
A) Программа жұмысын жалғастырады
B) Қате ескерілмейді
C) Программа тоқтайды
D) Жаңа айнымалы пайда болады
- finally: блогының қызметі қандай?
A) Тек қате шықса орындалады
B) Қате шықпаса орындалады
C) Қателерді болдырмайды
D) Қате шықса да, шықпаса да орындалады
- raise не істейді?
A) Айнымалыны жариялайды
B) Қате жібереді
C) Шарт қосады
D) Мән қайтарады
- assert қандай мақсатта қолданылады?
A) Цикл жасау
B) Қате тудыру
C) Тест жазу және шарт тексеру
D) Мәлімет енгізу
- Қайсысы Python-да жиі кездесетін қате?
A) NullError
B) TypeError
C) FindError
D) InsertError
- NameError дегеніміз не?
A) Айнымалы табылмағанда
B) Мәтін дұрыс болмағанда
C) Түсініксіз тип қолданғанда
D) Модуль қате болса
- Қолданушы анықтаған ерекше жағдай қалай жасалады?
A) error Exception(Exception): pass
B) def Error():
C) class MyError(Exception): pass
D) error = Exception
- Қате туралы хабарды басып шығару үшін не қолданылады?
A) print(error)
B) print(e)
C) except e:
D) except: print(e) •
- Жауаптар тізімі (61-80): • • • • • • • • • • • • A C B B C B A A C B • • • • • • • • • • B A C D B C B A C B • •

```
    •    Python-да модульді қалай импорттаймыз?
A) include math
B) import math
C) using math
D) module math
    •    Модульден тек бір функцияны импорттау:
A) import sqrt from math
B) from math import sqrt
```

```
C) math.sqrt = import
D) import(math.sqrt)
    •    import random не үшін қажет?
A) Қателерді өңдеу
B) Кездейсоқ сандар генерациясы
C) Модуль жою
D) Тізім сұрыптау
```

- Кітапхана дегеніміз не?
  A) Тек бір функция
  B) Айнымалылар жиыны
  C) Қате
  D) Функциялар мен модульдердің жинағы
- Модульді as кілтсөзімен қалай қолданамыз?
  A) import random as rnd
  B) import random is rnd
  C) random as import rnd
  D) using random → rnd
- dir(math) не үшін қажет?
  A) Математикалық есептер шығару
  B) Модульдің ішіндегі барлық функцияларды көру
  C) Модульді жою
  D) Модульді жаңарту
- help(random) не істейді?
  A) Модульді орнатады
  B) Модуль функциясын өшіреді
  C) Анықтама шығарады
  D) Мән жазады
- Қай модуль кездейсоқ бүтін сан береді?
  A) random
  B) math
  C) os
  D) time
- from random import randint деген не істейді?
  A) Модуль орнатады
  B) randint функциясын шақырады
  C) Файл ашады
  D) Мәтінді жояды
- Python-да сыртқы модуль орнату үшін қандай команда қолданылады?
  A) install module
  B) python get
  C) pip install
  D) setup run
- Python-да қандай цикл түрлері бар?
  A) for, while
  B) if, else
  C) run, stop
  D) open, close
- Python қандай бағдарламалау парадигмасын қолдайды?
  A) Тек объектіге бағытталған
  B) Тек процедуралық
  C) Процедуралық және объектіге бағытталған
  D) Тек функционалдық

- == және is арасындағы айырмашылық?
  A) Екеуі бірдей
  B) == мәнді, is сілтемені салыстырады
  C) == сілтемені, is мәнді салыстырады
  D) Екеуі тек сандарда қолданылады
- Python коды қандай кеңейтумен сақталады?
  A) .pt
  B) .py
  C) .txt
  D) .java
- Python 3-те қайсысы дұрыс print жазбасы?
  A) print "Hello"
  B) print: "Hello"
  C) print("Hello")
  D) echo("Hello")
- Қайсысы дұрыс функция синтаксисі?
  A) func my():
  B) def my():
  C) define my():
  D) function my():
- None дегеніміз не?
  A) Бос тізім
  B) 0
  C) Арнайы бос мән
  D) Қате
- Python коды қалай жазылады?
  A) Бөлшек-жолмен
  B) Фигуралық жақшамен
  C) Жол бағытымен
  D) Шегініс арқылы
- Python бағдарламасының алғашқы жолында қандай код жиі кездеседі?
  A) # This is python
  B) #!/usr/bin/python3
  C) // Python start
  D) begin python
- Python-да мәлімет типін өзгерту процесі қалай аталады?
  A) Data converting
  B) Type movement
  C) Type casting
  D) Variable assignment •
- Жауаптар тізімі (81-100): • • • • • • • • • • • • B B B D A B C A B C • • • • • • • • • • A C B B C B C D B C
- Python-да графикалық интерфейсті жасау үшін жиі қолданылатын кітапхана:
  A) NumPy
  B) Tkinter
  C) Requests
  D) JSON
- Tkinter дегеніміз не?
  A) Математикалық модуль
  B) Деректер базасы
  C) Графикалық интерфейс модулі
  D) Желілік модуль
- Графикамен жұмыс істеуге арналған модуль:
  A) math
  B) turtle

C) os
D) json

- pygame модулінің негізгі мақсаты:
A) Дерек жіберу
B) Ойын жасау
C) Қате өңдеу
D) Жолдарды бөлу
- turtle модулі не істейді?
A) Терезе ашады
B) Turtle графикасын салады
C) Файл жүктейді
D) Тест орындайды
- Tkinter терезесін бастау үшін қандай класс қолданылады?
A) tk.Screen()
B) tk.App()
C) tk.Tk()
D) tk.Gui()
- Tkinter-де батырма жасау үшін қолданылатын виджет:
A) Frame
B) Label
C) Button
D) Window
- mainloop() функциясы не үшін керек?
A) Экранды тазарту үшін
B) Бағдарлама тоқтату үшін
C) Интерфейсті жұмыс істетіп тұру үшін
D) Виджеттерді жою үшін
- Label элементінің міндеті:
A) Сурет салу
B) Мәтін шығару
C) Есептеу
D) Жаңа терезе ашу
- Entry элементі не үшін қолданылады?
A) Батырма жасау
B) Кескін салу
C) Мәтін енгізу
D) Диаграмма салу
- Pygame терезесін бастау үшін қолданылатын функция:
A) pygame.screen()
B) pygame.run()
C) pygame.display.set_mode()
D) pygame.show()
- Pygame-да оқиға (event) өңдеудің негізі:
A) event.trigger()
B) pygame.event.get()
C) input()
D) mouse.click()
- Pygame-да суретті экранға шығару үшін:
A) display.blit(image, coords)
B) draw.line()
C) render.image()
D) paint()
- Pygame-да кадр жиілігін басқару үшін:
A) pygame.FPS()

B) pygame.time.Clock()
C) pygame.loop()
D) pygame.frame()

- Кесінді сызу функциясы:
A) pygame.draw.rect()
B) pygame.line()
C) pygame.draw.line()
D) pygame.graph.line()

- matplotlib кітапханасының негізгі міндеті:
A) Желімен жұмыс
B) Графиктер салу
C) Қате өңдеу
D) Видео өңдеу

- Matplotlib-те график салу үшін ең жиі қолданылатын функция:
A) draw()
B) plot()
C) line()
D) show()

- Turtle модулінде түс орнату командасы:
A) set.color()
B) color()
C) turtle.color()
D) paint()

- Turtle-да экран тазарту үшін қолданылатын функция:
A) turtle.reset()
B) turtle.clear()
C) turtle.erase()
D) turtle.start()

- Turtle модулінде бағыт өзгерту командасы:
A) forward()
B) rotate()
C) turn()
D) right() •

- Жауаптар тізімі (101-120): • • • • • • • • • • •

B C B B B C C C B C

• • • • • • • • • •

C B A B C B B C B D

• •