

«ӘБІЛҚАС САҒЫНОВ АТЫНДАҒЫ ҚАРАҒАНДЫ ТЕХНИКАЛЫҚ
УНИВЕРСИТЕТІ» КОММЕРЦИЯЛЫҚ ЕМЕС АҚЦИОНЕРЛІК ҚОҒАМЫ

«Ақпараттық-есептеуіш жүйелер» кафедрасы

«Функционалдық және логикалық бағдарламалау» пәні бойынша

КУРСТЫҚ ЖҰМЫС

Тақырыбы: «Жиектерден секіретін DVD логотипі»

Бағасы _____

Орындаған: Зекен Алихан

BT-20-1 тобының студенті

Тексерген Искаков М.Б., доцент, т.ғ.к.
_____ «__» _____ 2023 г.

Комиссия мүшелері:

_____ Абильдаева Г.Б., АЕЖ кафедрасының аға оқытушысы

_____ Сүлеймен А.Е., АЕЖ кафедрасының аға оқытушысы

Қарағанды 2023

Әбілқас Сағынов атындағы Қарағанды техникалық университеті

АЕЖ кафедрасы

Бекітемін:

каф.меңгерушісі

Калинин А.А.

(қолы)

«_____» _____ 2023ж.

«Функционалдық және логикалық бағдарламалау» пәні бойынша
курстық жобаға тапсырма

Нұсқа №

СТУДЕНТ Зекен Алихан ТОП ВТ-20-1

Жоба тақырыбы: *Жиектерден секіретін DVD логотипі*

Бастапқы мәліметтер: Курстық жобаны орындауға арналған әдістемелік нұсқау.

Берілген нұсқаға сәйкес Python бағдарламалық тілінде «Жиектерден секіретін DVD логотипі» жобалау қажет. Орындау барысын, талдау нәтижелерін түсініктеме жазбада көрсету қажет.

1 А. Б Семенов. Проектирование и расчет структурированных кабельных систем и их компонентов. - М.:ДМК Пресс, 2019. – 416с.

2 Смагулова А.С. Желілік технологиялар: оқулық. Қарағанды: ҚарМТУ, 2018.

Тапсырманың берілген күні 12.09.2023ж., жобаны қорғау күні 11.12.2023ж.

Жоба жетекшісі Искаков М.Б.

(күні, қолы)

Тапсырманы орындаған Зекен Алихан

(күні, студентің қолы)

Мазмұны

Кіріспе.....	3
1 Python тілінде бағдарламалау негіздері	4
1.1 Python тілінің ерекшелігі	5
1.2 Python бағдарламасының құрамы.....	7
2 Python тіліндегі бағдарлама мәтінін талдау	12
2.1. Бағдарлама мәтіні бойынша алгоритмнің вербальдық сипаттамасы	12
2.2 Бағдарлама алгоритмінің блок схемасын жасау	14
2.3 Бағдарламада қолданылған операторлардың сипаттамасы	16
Қорытынды.....	17
Қолданылған әдебиеттер тізімі	19

Кіріспе

Python-әзірлеушінің өнімділігін, кодтың оқылуын және оның сапасын жақсартуға, сондай-ақ онда жазылған бағдарламалардың тасымалдануын қамтамасыз етуге бағытталған динамикалық қатаң теру және жадты автоматты басқарумен жоғары деңгейлі жалпы мақсаттағы бағдарламалау тілі. Тіл толығымен объектіге бағытталған, өйткені бәрі объект болып табылады. Тілдің ерекше ерекшелігі-код блоктарын бос орындармен бөлу. Тіл ядросының синтаксисі минималистік болып табылады, соның арқасында іс жүзінде құжаттамаға жүгіну сирек кездеседі.

Заманауи бағдарламалау тілдері әзірлеушілерге қызықты бағдарламалар жасауға кең мүмкіндіктер береді. Осы курстық жұмыстың аясында біз Python бағдарламалау тілін қолдана отырып, DVD логотипінің анимациясын жасауды қарастырамыз.

Python-қарапайым синтаксисті және әртүрлі тапсырмалар үшін көптеген кітапханаларды қамтамасыз ететін әмбебап тіл. Біз анимация негіздерін, терминал жұмысын және Python бағдарламасының құрылымын үйренуге мүмкіндік беретін ойын-сауық жобасының мысалы ретінде DVD логотипінің анимациясын жасауды таңдадық.

Бұл курстық жұмыстың мақсаты-терминалда секіретін DVD логотипінің анимациясын жасайтын Python бағдарламасын жасау. Біз терминалдағы шығуды басқару және визуалды эффект жасау үшін `bext` кітапханасын қолданамыз. Жоба Python - да бағдарламалау негіздерін үйренуге мүмкіндік беріп қана қоймай, сонымен қатар анимация және терминалмен өзара әрекеттесу принциптерін қарастыруға мүмкіндік береді.

1 Python тілінде бағдарламалау негіздері

Python-Гидо ван Россум жасаған жоғары деңгейлі, интерпретацияланған бағдарламалау тілі. Оның дизайн философиясы кодтың оқылуына және синтаксистің анықтығына назар аударады, бұл Python жаңадан бастаушылар мен тәжірибелі әзірлеушілер үшін әртүрлі тапсырмалар үшін тамаша таңдау жасайды.

Python-дың негізгі ерекшеліктеріне динамикалық теру кіреді, бұл деректердің икемді жұмыс істеуіне мүмкіндік береді және модульдік және оңай кеңейтілетін кодты құруға ықпал ететін объектіге бағытталған бағдарламалау. Оның интерпретациясын атап өту маңызды, өйткені ол компиляция қадамдарын азайту арқылы тестілеу мен әзірлеуді жеңілдетеді.

Python-да желілермен жұмыс істеуден бастап деректерді өңдеуге дейінгі әртүрлі тапсырмаларға арналған құралдар мен модульдердің кең ауқымын қамтитын кең стандартты кітапхана бар. Бұл әзірлеушілердің жұмысына ықпал етеді және кодты "нөлден" жазу қажеттілігін азайтады.

Python сонымен қатар кросс-платформа болып табылады, яғни Python-да жазылған бағдарламалар әртүрлі операциялық жүйелерде өзгеріссіз орындалуы мүмкін. Бұл тілге әмбебаптық пен қол жетімділік береді.

Веб-әзірлеуде, деректер ғылымында, жасанды интеллектте және басқа салаларда кеңінен қолданыла отырып, Python заманауи бағдарламашының арсеналында маңызды құралға айналады. Ашық көзі, белсенді қауымдастығы және егжей-тегжейлі құжаттамасы бар Python әлемдегі ең танымал бағдарламалау тілдерінің бірі болып қала береді. Python-әзірлеушінің өнімділігі мен кодты оқуға бағытталған жоғары деңгейлі жалпы мақсаттағы бағдарламалау тілі. Python ядросының синтаксисі минималистік. Сонымен қатар, стандартты кітапхана көптеген пайдалы функцияларды қамтиды. Тіл айқын және дәйекті синтаксиске, ойластырылған модульділікке және масштабталуға ие, бұл Python-да жазылған бағдарламалардың бастапқы кодын оңай оқуға мүмкіндік береді.

Python-белсенді дамып келе жатқан бағдарламалау тілі, тілдік қасиеттерді қосатын және өзгертетін жаңа нұсқалар шамамен екі жарым жылда бір рет шығады. Ол адам қызметінің көптеген салаларында қолданылады.

Python-ең "жас" бағдарламалау тілі емес, бірақ ол тым ескі емес. Ол құрылған кезде "Паскаль" немесе "Си" сияқты тілдер болған. Сондықтан, "питонды" құру кезінде авторлар әзірлеушілер үшін әртүрлі платформалардың ең жақсысын алуға тырысты. Шын мәнінде, Python-бұл 8-ден астам түрлі тілдердің сәтті шешімдерінің "джемі"[1].

Питон барлық жалпы операциялық жүйелерді қолдайды. Ол қалта компьютерлерінде де, үлкен серверлерде де жақсы жұмыс істей алады. Егер платформа айтарлықтай ескірген болса, ол ядро тірегінен шығарылады. Мысалы, тілдің 2.6-дан басталатын нұсқалары енді Windows 95, 98 және ME платформаларымен жұмыс істемейді. Қажет болса, тілдің заманауи құралдарын қолданудан бас тарту арқылы ескі нұсқаларды қолдануға болады. Содан кейін қосымша, соның ішінде осы ОЖ-мен жұмыс істейді.

Python бағдарламалық өнімдерді әзірлеу нарығындағы көптеген рейтингтер мен ұсыныстарды талдаумен дәлелденген ең танымал және танымал бағдарламалау тілдеріне жатады. Бұл өте қарапайым, сондықтан тілді үйрену көп уақытты қажет етпейді.

Python іске қосылған кезде pythonshell аудармашысының терезесі пайда болады. Ол адамға түсінікті кодты машина тіліне, яғни құрылғының процессоры орындай алатын кодқа аударады. Мұнда "Файл", "өңдеу", "күйін келтіру", "опциялар", "терезе", "Анықтама"қойындылары орналасқан. Бағдарламаларды құру үшін "Файл" қойындысына өтіп, жаңа файл жасау керек. Біздің алдымызда кодты жазатын терезе ашылады. Бағдарламаны жазғаннан кейін ол PythonShell-де орындалады[9].

Бағдарламаларды құру үшін көбінесе қосымша мүмкіндіктер қажет. Ол үшін арнайы кітапханалар бар. Кітапханаларды батырмалары, суреттері және т.б. бар терезе қосымшаларын жасау үшін пайдалануға болады. Ойындар жасауға арналған арнайы кітапханалар бар. Олардың кейбіреулері Python-да орнатылған, кейбіреулерін бөлек жүктеу керек.

1.1 Python тілінің ерекшелігі

Python – жоғары деңгейлі, түсіндірілетін және жалпы мақсаттағы бағдарламалау тілі. Ол кодты оқуға және қарапайым синтаксиске баса назар аудара отырып, оны жаңадан бастағандар үшін өте қолжетімді етеді, сонымен қатар тәжірибелі әзірлеушілер үшін тиімділік пен икемділікті қамтамасыз етеді. Мұнда Python тілінің кейбір негізгі мүмкіндіктері берілген:

Динамикалық Теру:

- Python-да айнымалылар декларация процесінде белгілі бір деректер түрімен байланысты емес. Деректер түрі бағдарламаны орындау кезінде автоматты түрде анықталады, бұл икемділік пен әзірлеуге ыңғайлылық береді.

Объектіге бағытталған бағдарламалау (ООП):

- Python - бұл толығымен объектіге бағытталған бағдарламалау тілі, яғни Python-дағы барлық нәрсе объект болып табылады және код сыныптар мен нысандар түрінде ұйымдастырылады.

Интерпретация:

- Python-түсіндірілетін тіл. Кодты интерпретатор алдын-ала құрастырусыз қатар-қатар орындайды. Бұл тестілеуді, жөндеуді жеңілдетеді және жедел кері байланыс береді.

Синтаксистің қарапайымдылығы және оқылуы:

- Python синтаксисі кодтың оқылуына баса назар аудара отырып жасалған. Минималистік конструкциялар және бұйра қапсырмалардың орнына шегіністерді қолдану кодты оқылымды және сүйемелдеуді жеңілдетеді.

Икемділік және динамизм:

- Python кодты жазу кезінде икемді опцияларды ұсынады. Барлық айнымалылар динамикалық болып табылады, бұл бағдарламаны орындау барысында айнымалылардың деректер түрін оңай өзгертуге мүмкіндік береді.

Модульдік және портативтілік:

- Python модульдікті қолдайды, бұл қайта пайдалануға болатын компоненттерді жасауға мүмкіндік береді[1]. Сонымен қатар, Python-да жазылған код әртүрлі операциялық жүйелер арасында оңай тасымалданады.

Keң Стандартты Кітапхана:

- Python әртүрлі тапсырмаларға арналған көптеген модульдер мен құралдарды қамтитын кең стандартты кітапханамен бірге келеді. Бұл кодты "нөлден" жазу қажеттілігін азайтады және әзірлеушілердің жұмысын жақсартады.

Жадты Басқару:

- Python жадты қоқыс жинағышпен автоматты түрде басқарады. Бұл әзірлеушіге жадты бөлу және босату туралы алаңдамаудың қажеті жоқ дегенді білдіреді.

Динамикалық Импорттау:

- Python модульдерді динамикалық импорттауды қолдайды, бұл бағдарламаларға әр түрлі жұмыс жағдайларына бейімделуге және қажет болған жағдайда модульдерді жүктеуге мүмкіндік береді.

Ерекшеліктерді Өңдеу:

- Python бағдарламадағы қателер мен күтпеген жағдайларды басқаруды жеңілдететін ерекше жағдайларды өңдеу механизмін ұсынады.

Қолданудың Кең Мүмкіндіктері:

- Python веб-әзірлеу, ғылыми зерттеулер, жасанды интеллект, деректерді талдау, автоматтандыру, ойын әзірлеу және т. б. сияқты әртүрлі салаларда қолданылады[2].

Белсенді қоғамдастық және қолдау:

- Python-да қолдауды, тәжірибе алмасуды және тілді дамытуды қамтамасыз ететін белсенді және кең әзірлеушілер қауымдастығы бар.

Python-бұл әзірлеушілер арасында танымал ететін бірқатар ерекше ерекшеліктері бар бағдарламалау тілі. Динамикалық теру деректермен икемді жұмыс істеуге мүмкіндік береді, ал объектіге бағытталған бағдарламалауды толық қолдау кодтың құрылымдық ұйымдастырылуын қамтамасыз етеді. Интерпретация даму процесін икемді және жылдам етеді.

Қарапайым синтаксис және оқуға баса назар аудару көптеген әзірлеушілер үшін қол жетімді етті. Тілдің икемділігі мен динамикасы әртүрлі қосымшаларды жасауға мүмкіндік береді, ал модульдік қайта қолданылатын кодты жасауға ықпал етеді.

Кең және бай функционалдығы бар стандартты Python кітапханасы әзірлеушілердің өнімділігін арттыра отырып, кодты жазуды "нөлден" азайтады. Жадты басқару және ерекше жағдайларды өңдеу дамуды тұрақты және қауіпсіз етеді.

Python веб-әзірлеуден бастап ғылыми зерттеулерге дейін әртүрлі салаларда қолданылады, бұл оның әмбебаптығын көрсетеді. Белсенді әзірлеушілер

қауымдастығы тілді қолдауға, тәжірибе алмасуға және одан әрі дамытуға ықпал етеді. Python өзінің қуаты, пайдаланудың қарапайымдылығы және мүмкіндіктердің байлығы арқасында көптеген бағдарламашылар үшін таңдау тілі болып қала береді.

1.2 Python бағдарламасының құрамы

Python бағдарламасының құрамына әр түрлі компоненттер кіреді, олардың әрқайсысы белгілі бір функцияларды орындайды. Мұнда Python бағдарламасының негізгі құрамы берілген:

1. Кітапхананы импорттау-бұл тілдің стандартты мүмкіндіктерін кеңейтетін сыртқы модульдерді немесе пакеттерді қосу механизмі[3]. Бұл бағдарламашыларға белгілі бір мәселелерді шешу үшін басқа әзірлеушілер жасаған дайын шешімдерді пайдалануға мүмкіндік береді.

Түрлері:

- Модульдерді импорттау: Python-да модуль-бұл басқа бағдарламаларда қолдануға болатын код пен анықтамаларды қамтитын файл. Модульді импорттау импорт кілт сөзін, содан кейін модуль атауын қолдану арқылы жүзеге асырылады.

- Функцияларды/сыныптарды импорттау: тұтас модульді импорттаудан басқа, кодта пайдалану үшін модульден белгілі бір функцияларды немесе сыныптарды импорттауға болады. Мұны from кілт сөзімен жасауға болады.

- Имен кеңістігі: импортталған элементтер бағдарламада өздерінің имен кеңістігінде қол жетімді болады, бұл бағдарламаның әртүрлі бөліктері арасындағы имен қайшылықтарын болдырмайды.

- Стандартты кітапхана: Python математика, файлдармен жұмыс, желі және т.б. сияқты әртүрлі тапсырмаларға арналған көптеген модульдерді қамтитын кең стандартты кітапханамен бірге келеді.

- Тәуелділікті басқару: бір сияқты пакетті басқару құралымен орнатылған үшінші тарап кітапханаларын пайдалану жобандағы тәуелділіктер мен жаңартуларды тиімді басқаруға мүмкіндік береді.

Кітапханаларды импорттау Python-да дамудың негізгі бөлігі болып табылады, бұл дайын шешімдерді қолдануға және бағдарламашылардың тиімділігін арттыруға мүмкіндік береді[4].

2. Айнымалылар-деректерді жақта сақтау үшін қолданылатын аталған контейнерлер. Бағдарламалауда айнымалылар шешуші рөл атқарады, бұл бағдарламашыларға бағдарламаны орындау барысында мәндерді өзгертуге және өзгертуге мүмкіндік береді.

Сипаттама:

- Айнымалы атау: айнымалы әріптен немесе астын сыздан басталуы керек бірегей атаумен анықталады, одан кейін әріптер, сандар немесе басқа астын сызу болуы мүмкін.

- Мән тағайындау: айнымалыға = операторының көмегімен мән беріледі. Мәнді тағайындау, егер ол бұрын болмаған болса, айнымалыны құру болып табылады.

- Динамикалық теру: Python-да оны жасау кезінде айнымалы түрін көрсетудің қажеті жоқ. Деректер түрі мән берілген кезде автоматты түрде анықталады.

- Деректер түрлері: айнымалылар бүтін сандар, өзгермелі нүкте сандары, жолдар, тізімдер, сөздіктер және т.б. сияқты әртүрлі деректер түрлерін қамтуы мүмкін.

- Мәндерді өзгерту: айнымалының мәнін бағдарламаны орындау барысында өзгертуге болады, бұл айнымалыларды динамикалық және икемді етеді.

- Айнымалыларды өрнектерде қолдану: айнымалыларды әр түрлі операцияларды орындау үшін өрнектерде қолдануға болады, бұл деректердің манипуляциясын қамтамасыз етеді.

- Әмбебаптық және қайта пайдалану: айнымалылар әмбебап және қайта пайдалануға болатын кодты жасауға мүмкіндік береді және оларды пайдалану бағдарламаларды оқылымды және түсінікті етеді.

- Жадты босату: Python жадты автоматты түрде басқарады, ол енді пайдаланылмаған кезде айнымалылар алатын орынды босатады.

Айнымалылар бағдарламалаудағы негізгі құрал болып табылады, деректермен жұмыс істеу құралын қамтамасыз етеді және қолданбаларды құру процесінде динамизм мен икемділікті қамтамасыз етеді.

3. Енгізу және шығару операторлары-бағдарламаның пайдаланушымен өзара әрекеттесуін және жұмыс нәтижелерін шығаруды қамтамасыз етеді. Бұл операторлар деректермен жұмыс істеудің ыңғайлы құралдарын ұсынады.

Деректерді Енгізу (енгізу): енгізу операторы пайдаланушыдан кіріс алу үшін қолданылады. Бағдарлама пернетақтадан деректерді енгізуді күтіп тоқтайды. Енгізілген деректерді кейінірек пайдалану үшін айнымалыға сақтауға болады[6].

Деректерді шығару (print): нәтижелерді экранда көрсету үшін Print операторы қолданылады. Бұл мәтін, айнымалы мәндер, есептеу нәтижелері немесе көрсетуге арналған кез келген басқа ақпарат болуы мүмкін.

Пішімделген Шығыс: Print операторы деректерді белгілі бір форматта ыңғайлы шығаруға мүмкіндік беретін пішімделген шығуды қолдайды.

Жолдың бөлінуі және аяқталуы: print операторында шығатын мәндер мен жолдың аяқталуы арасындағы бөлуді басқару үшін параметрлерді пайдалануға болады.

Арнайы таңбаларды қолдану: шығарылымның орнын өзгерту үшін \n (Жаңа жол) немесе \t (қойынды) сияқты арнайы таңбаларды пайдалануға болады.

Файлдармен жұмыс: Python бағдарламаларға сыртқы файл ресурстарымен байланысуға мүмкіндік беретін файлдармен (open, read, write) жұмыс істеу үшін Енгізу/шығару операторларын ұсынады.

Енгізу/шығару операторлары бағдарламаның пайдаланушымен өзара әрекеттесуінің және ақпараттық Шығыс құрудың негізгі элементі болып табылады, бұл оларды Python бағдарламалаудың маңызды бөлігіне айналдырады.

4. Python-да ерекшеліктерді өңдеу бағдарламаға қателіктер мен күтпеген жағдайларға дұрыс жауап беруге мүмкіндік береді, бұл апаттық жұмыстың аяқталуына жол бермейді.

Сипаттама:

- Try операторы: ерекше жағдайлар туындауы мүмкін код блогын жақтау үшін қолданылады.

- Except операторы: ерекшелік болған жағдайда орындалатын код блогын анықтайды. Белгілі бір ерекшелікті өңдеуге немесе кез келген ерекшелікті өңдеуге мүмкіндік береді.

- Else операторы: егер try блогында ерекшеліктер болмаса, орындалатын қосымша блок[7].

- Finally операторы: ерекшелік пайда болды ма, жоқ па, әрқашан орындалатын қосымша блок. Ресурстарды босату үшін жиі қолданылады.

5. Python-дағы шартты операторлар бағдарламаға шарттарға байланысты шешімдер қабылдауға және әртүрлі әрекеттерді орындауға мүмкіндік береді.

Сипаттама:

- If операторы: егер берілген шарт ақиқат болса, код блогын орындау үшін қолданылады.

- Elif операторы: алдыңғы шарттар орындалмаса, тексеру үшін қосымша шарттар қосуға мүмкіндік береді.

- Else операторы: алдыңғы шарттардың ешқайсысы дұрыс болмаса, орындалатын код блогын анықтайды.

6. Python-дағы циклдар белгілі бір шарттарға байланысты Код блогын бірнеше рет орындау үшін қолданылады.

Сипаттама:

- For циклі: реттілік бойынша Итерация үшін қолданылады (мысалы, тізім) немесе басқа қайталанатын объект.

- While циклі: берілген шарт ақиқат болғанша орындалады. Итерация саны алдын-ала белгісіз болған кезде қолайлы.

- Break және continue операторлары: циклдің орындалуын басқару үшін қолданылады. үзіліс циклды тоқтатады, жалғасы келесі итерацияға өтеді.

7. Python-дағы функциялар кодты жеке блоктарға ұйымдастыру механизмін ұсынады, бұл оны оқуға және қайта пайдалануға мүмкіндік береді.

Сипаттама:

- Функцияның анықтамасы (def): функцияны құру үшін def операторы қолданылады. Функция параметрлерді қабылдай алады және мәндерді қайтара алады.

- Айнымалылардың ауқымы: функция ішіндегі айнымалылардың Ғаламдық доменнен бөлек көру аймағы болады.

Ерекшеліктерді өңдеу, шартты операторлар, циклдар мен функциялар Python тілінің негізгі элементтері болып табылады, бұл оның икемділігі мен ыңғайлылығын қамтамасыз етеді[8].

Бұл негізгі компоненттер Python бағдарламаларының негізін құрайды, ал әр түрлі кітапханалар мен құрылымдар бағдарламалаудың әр түрлі бағыттары үшін тілдің функционалдығын кеңейтеді.

1.3 Интегралды бағдарламалау ортасының қысқа сипаттамасы

Интеграцияланған даму ортасы (ide) – бұл бағдарламашыларға бағдарламалық кодты тиімді жасауға көмектесетін бағдарламалық жасақтама. Ол бағдарламалық жасақтаманы өңдеу, құру, тестілеу және орау сияқты мүмкіндіктерді пайдалану оңай қолданбаға біріктіру арқылы әзірлеушілердің өнімділігін арттырады. Жазушылар мәтіндік редакторларды, ал бухгалтерлер электрондық кестелерді қолданатыны сияқты, бағдарламалық жасақтама жасаушылар да жұмысты жеңілдету үшін IDE қолданады.

Кодты жазу үшін кез-келген мәтіндік редакторды қолдануға болады. Алайда, интеграцияланған даму орталарының көпшілігі (ide) мәтінді өңдеуден тыс мүмкіндіктерді қамтиды. Олар бағдарламалық жасақтаманы әзірлеу процесін әлдеқайда тиімді ете отырып, жалпы әзірлеуші құралдарына орталық интерфейсті ұсынады. Әзірлеушілер әртүрлі бағдарламалық жасақтаманы қолмен біріктірудің және конфигурациялаудың орнына жаңа қосымшаларды бағдарламалауды тез бастай алады. Сонымен қатар, олар барлық құралдарды үйренудің қажеті жоқ, тек бір қолданбаға назар аудара алады. Төменде әзірлеушілердің IDE қолдануының кейбір себептері келтірілген:

– Бағдарламалау тілдерінде мәлімдемелерді құрылымдау ережелері бар. IDE бұл ережелерді білетіндіктен, ол бастапқы кодты автоматты түрде жазуға немесе өңдеуге арналған көптеген ақылды мүмкіндіктерді қамтиды.

– Ide кейбір сөздерді автоматты түрде қою немесе курсивпен немесе әртүрлі қаріп түстерін қолдану арқылы жазылған мәтінді пішімдей алады. Бұл көрнекі белгілер бастапқы кодты оқылымды етеді және кездейсоқ синтаксистік қателер туралы бірден хабарлайды[2].

– Іздеу жүйесінде сөздерді тере бастағанда, әртүрлі іздеу сұраулары пайда болады. Сол сияқты, ide әзірлеуші тере бастаған кезде кодты енгізуді аяқтауды ұсынуы мүмкін.

– Кодты қайта өңдеу-бұл бастапқы функционалдығын өзгертпестен оны тиімдірек және оқылымды ету мақсатында бастапқы кодты қайта құрылымдау процесі. Ide авторефакторингті белгілі бір дәрежеде қолдана алады, бұл әзірлеушілерге кодты тез және оңай жақсартуға мүмкіндік береді. Команданың басқа мүшелері оқылатын кодты тезірек түсінеді, бұл топ ішіндегі ынтымақтастыққа ықпал етеді.

– IDE кодты өзгертудің әр бөлігі болып табылатын қайталанатын әзірлеу тапсырмаларын орындау арқылы бағдарламашылардың өнімділігін арттырады. Төменде ide орындайтын тұрақты кодтау тапсырмаларының мысалдары келтірілген.

– IDE кодты амалдық жүйеге түсінікті жеңілдетілген тілге құрастырады немесе түрлендіреді. Кейбір бағдарламалау тілдері "дәл уақытында" компиляциясын жүзеге асырады, онда ide оқылған кезде түсінікті кодты қолданба ішіндегі машиналық кодқа түрлендіреді.

– Ide әзірлеушілерге бағдарламалық жасақтама басқа әзірлеушілердің кодымен біріктірілмес бұрын және күрделі интеграциялық тесттер іске қосылмас бұрын модульдік тестілерді жергілікті түрде автоматтандыруға мүмкіндік береді.

– Жөндеу-бұл тестілеу анықтаған кез келген қателерді немесе кемшіліктерді түзету процесі. Отладтау мақсатында IDE-дің ең үлкен құндылықтарының бірі-сіз кодты жолда қарап, оны орындаған кезде және кодтың әрекетін тексере аласыз. Сондай-ақ, ide-де әзірлеуші теріп жатқан кезде де нақты уақыт режимінде адам қателіктерінен туындаған ақауларды анықтайтын бірнеше жөндеу құралдары бар[1].

Интеграцияланған даму орталары (ide) қандай қолданбаларды әзірлеуді қолдайтынына және қалай жұмыс істейтініне байланысты бірнеше түрлі санаттарға бөлуге болады. Дегенмен, көптеген ide бағдарламалық жасақтамаларын бірнеше санатқа жатқызуға болады. Төменде IDE кейбір түрлері берілген.

Жергілікті ide әзірлеушілері жергілікті ide-ді тікелей жергілікті машиналарға орнатады және іске қосады. Олар сондай-ақ кодтау қалауларына, жоба талаптарына және даму тіліне байланысты әртүрлі қосымша кітапханаларды жүктеп алып, орнатуы керек. Жергілікті IDE-лер теңшелетін және орнатылғаннан кейін Интернет байланысын қажет етпесе де, олар бірнеше мәселелермен бірге келеді.

Бұлтты ide әзірлеушілері бұлтты ide-ді кодты тікелей шолғышта жазу, өңдеу және құрастыру үшін пайдаланады, бұл оларды бағдарламалық жасақтаманы жергілікті машиналарға жүктеу қажеттілігінен құтқарады. Бұлтты IDE дәстүрлі IDE-ге қарағанда бірқатар артықшылықтарға ие. Олардың кейбіреулері төменде көрсетілген.

Стандартталған даму ортасы-бағдарламалық жасақтама топтары стандартты даму ортасын құру үшін бұлтқа негізделген IDE-ді орталықтан теңшей алады. Бұл әдіс оларға жергілікті машинаның конфигурациясындағы айырмашылықтарға байланысты туындауы мүмкін қателіктерден аулақ болуға көмектеседі.

2 Python тіліндегі бағдарлама мәтінін талдау

2.1. Бағдарлама мәтіні бойынша алгоритмнің вербальдық сипаттамасы

Курстық жұмыс тапсырмасы DVD логотипінің терминал бойымен оның шетінен секірулермен қозғалысын имитациялайтын қарапайым анимация жасау. Бағдарлама әртүрлі түстердің бірнеше логотиптерін жасайды және оларды кездейсоқ экранда орналастырады. Содан кейін логотиптер әр бағытта қозғалады, терминалдың шетінен секіреді, әр секірген сайын түсі өзгереді.

Кодтың маңызды элементтері:

- Логотиптер түс, координаттар және қозғалыс бағыты туралы ақпаратты қамтитын ассоциативті массивтер түрінде ұсынылған.

- Бағдарламаның негізгі циклі әр логотиптің қозғалысын, оның терминал жиектерімен соқтығысуын, секірулерін және түсінің өзгеруін өңдейді.

- Бағдарлама терминалдағы түстермен жұмыс істеу үшін `next` модулін пайдаланады.

- Терминалдың бұрыштарынан секіру саны туралы ақпарат көрсетіледі.

Бұл код қарапайым анимацияның мысалы болып табылады және оны терминал жұмысын үйрену, кездейсоқ сандарды құру, анимация және Python-да мәліметтер құрылымын құру үшін оқу материалы ретінде пайдалануға болады.

Бағдарлама қолданыста

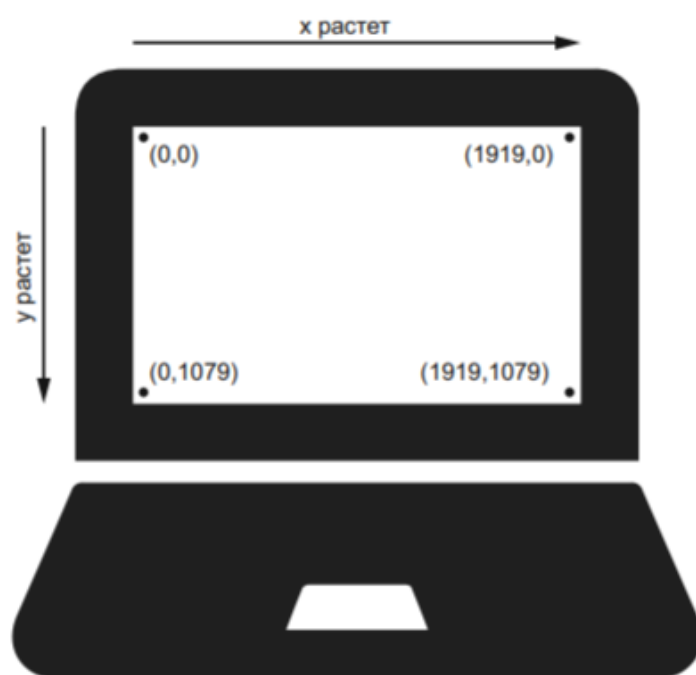
Орындалу нәтижесі `bouncingdvd.py` келесідей көрінеді (сурет-1).



Сурет-1 Бағдарламада диагональ бойынша қозғалатын DVD логотиптері `bouncingdvd.py`

Жұмыс сипаттамасы

Бағдарламалауда x координаты математикадағыдай нысанның көлденең орналасуына, ал y координаты тігінен сәйкес келеді. Алайда, бастап математикадан бетпе-бет, бастау нүктесі $(0, 0)$ экранның жоғарғы сол жақ бұрышында орналасқан және y координаты төмендеген сайын өседі. X координаты математикадағыдай объект оңға қарай жылжыған сайын өседі. Суретте. 5.2 экранның координаттар жүйесі көрсетілген. Best Модулінің `goto()` функциясы келесіге қоңырау шалған кезде өзін осылай ұстайды. `goto (0, 0)` мәтін енгізу курсоры терминал терезесінің жоғарғы сол жақ нүктесіне орналастырылған. Әрбір шағылысқан логотип біздің бағдарламада 'color', 'direction', 'x' және 'y' кілттері бар ассоциативті массив арқылы ұсынылған. 'X' және 'y' бүтін мәндері экрандағы логотиптің орнын көрсетеді. Бұл мәндер келесі функцияға берілгендіктен. `goto ()`, олардың үлкейтуі логотипті оңға және төмен жылжытады, ал кішірейту солға және жоғары жылжытады (сурет-2).



Сурет-2 Бастау нүктесі $(0, 0)$ экранның жоғарғы сол жақ бұрышында орналасқан және X және y координаттары сәйкесінше оңға және төмен жылжыған сайын өсуі

Код сипаттамасы

Бұл код терминалдың шеттерінен секіріп, экранда DVD логотипінің қозғалысын имитациялайтын бағдарлама. Міне, кодтың қысқаша сипаттамасы:

Кітапханаларды импорттау:

- `sys`, `random`, `time` жүйелік функциялармен жұмыс істеу, кездейсоқ сандарды құру және уақытты басқару үшін қолданылады.

- `best`-терминалдағы түстермен жұмыс істеуге арналған кітапхана. Алайда, егер ол болмаса, бағдарлама орнату қажеттілігі туралы хабарлама шығарады.

Тұрақтылар:

- `WIDTH` және `HEIGHT`-терминал өлшемдері.

- Number_of_logos-экрандағы логотиптер саны.
- PAUSE_AMOUNT-кадрлар арасындағы кідіріс (логотиптің қозғалу жылдамдығын анықтайды).
- Түстер-логотиптерге арналған түстер тізімі.
- Up_right, UP_LEFT, DOWN_RIGHT, DOWN_LEFT - логотиптің қозғалыс бағыттары.

Логотиптерді құру:

- Логотиптер кездейсоқ түспен, координаттармен және қозғалыс бағытымен жасалады.

Негізгі цикл:

- Терминалдың шетінен логотиптердің қозғалысы мен серпілістерін өңдейтін шексіз цикл.

- Циклдің әр итерациясында логотиптер олардың ағымдағы бағыты мен түсін ескере отырып, жаңа жерде көрсетіледі.

- Экранда бұрыштардан секіру саны көрсетіледі.

Терминал жиектерімен соқтығысуды бақылау:

- Егер логотип бұрышқа соқтығысса, оның бағыты өзгереді және секіру есептегіші артады.

Түс өзгерген кезде түс өзгереді:

- Логотип терминалдың шетінен әр секірген сайын оның түсі өзгереді.

Ақпаратты шығару және сурет салу:

- Бұрыштардан секіру саны көрсетіледі.
- Логотиптер терминалда олардың түсі мен координаттарын ескере отырып көрсетіледі.

Үзілісті жою:

- Бағдарлама keyboardinterrupt (Ctrl+C) ерекшелігін ұстайды және оның орындалуын аяқтайды.

Бағдарламаны іске қосу:

- Егер код тәуелсіз сценарий ретінде іске қосылса, main () функциясы шақырылады.

Бұл код терминалда түстерді қолдана отырып, DVD логотипі үшін қарапайым визуализация жасайды және шеттермен соқтығысқан кезде бағытты өзгертеді.

2.2 Бағдарлама алгоритмінің блок схемасын жасау

Блок-схема-бұл мәселені шешудің қадамдарын көрсететін диаграмма. Келесі блок-схемаларды бөліктері үшін толығырақ сипаттама :

Бағдарламаның басталуы:

- Блок-схеманың басталуы бастапқы блокпен белгіленеді.

Вехт кітапханасын тексеру:

- Шарты бар Блок: vехt кітапханасының болуы тексеріледі.
- Егер кітапхана жоқ болса, хабарлама шығады және бағдарлама орындалуын аяқтайды.

Тұрақтыларды инициализациялау:

- Терминалдың ені мен биіктігі, логотиптер саны, кідіріс және түстер сияқты тұрақтылар орнатылады.

Логотиптерді құру:

- Кездейсоқ сипаттамалары бар логотиптер жасауға арналған Цикл (түс, координаттар, бағыт).

- Бұрыштың соқтығысуын болдырмау үшін X координатының жұп екеніне кепілдік беріледі.

Негізгі цикл:

- Логотиптердің қозғалысы мен серпілістерін өңдеуге арналған шексіз цикл.

Әр логотипті өңдеу:

- Логотиптер тізіміндегі әрбір логотипке арналған Цикл.

- Терминалдағы алдыңғы логотиптің орнын тазарту.

Бұрыштардан секірулерді тексеру:

- Логотиптің терминалдың бұрышына соқтығысқанын тексеріңіз.

- Логотиптің бағытын өзгерту және секіру есептегішін ұлғайту.

Шеттерден секірулерді тексеру:

- Терминалдың сол, оң, жоғарғы және төменгі жиектерімен соқтығысуды тексеру.

- Ережелерге сәйкес логотиптің бағытын өзгерту.

Түс өзгерген кезде түс өзгереді:

- Егер логотип қайта оралса, оның түсі өзгереді.

Логотипті жылжыту:

- Логотиптің координаттарын оның бағытына сәйкес өзгерту.

Ақпаратты көрсету:

- Бұрыштардан секіру саны туралы ақпаратты шығару.

Логотиптерді салу:

- Терминалдағы жаңа орындарында логотиптерді көрсету.

- Кешіктіру және экранды жаңарту:

- Кадрлар арасындағы кідіріс және экранды жаңарту.

Үзілісті өңдеу:

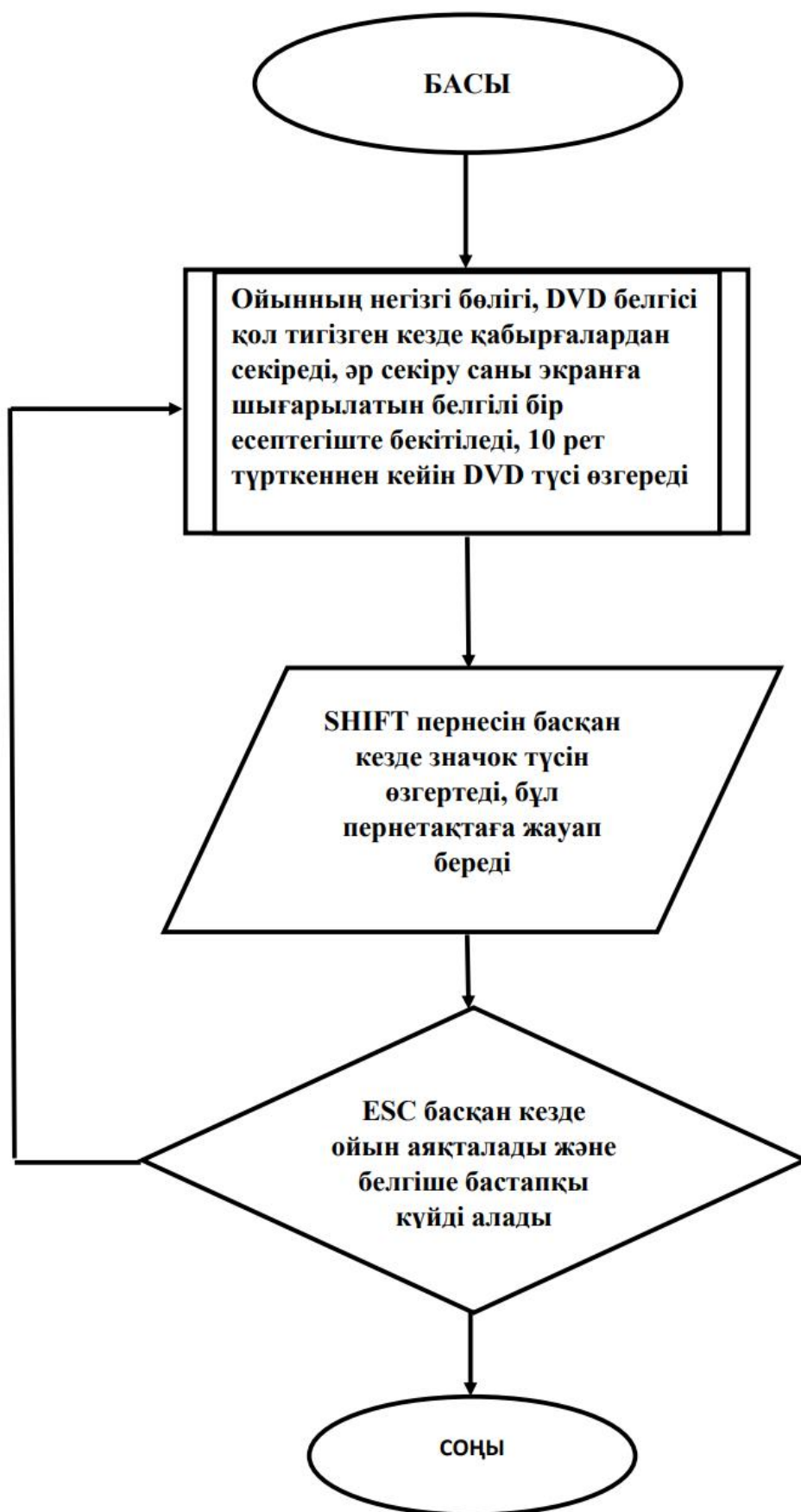
- Бағдарламаның орындалуын үзу кезінде ерекшелікті өңдеу (Ctrl+C).

- Хабарды шығару және бағдарламаны аяқтауды аяқтау.

Бағдарламаның соңы:

- Блок-схеманың соңы "соңы" блогымен белгіленеді.

Бұл бағдарлама алгоритмінің жалпы мәтіндік сипаттамасы, Бұл бағдарлама орындайтын әрекеттер тізбегі.



Блок-схема-1 Жиектерден секіретін DVD логотипі

2.3 Бағдарламада қолданылған операторлардың сипаттамасы

Бағдарлама логиканы басқару және әрекеттерді орындау үшін әртүрлі операторларды пайдаланады. Міне, олардың қысқаша сипаттамасы:

Import операторы:

- Бағдарламаны орындау үшін қажетті кітапханалар мен модульдерді импорттау үшін қолданылады. Мысалы, `Import sys, random, time`.

Try-except операторы:

- Ерекшеліктерді өңдейді. Бұл жағдайда, егер `bext` кітапханасы болмаса, `importerror` өңдеу үшін қолданылады.

Тағайындау операторлары (=):

- Деректер құрылымдарының айнымалылары мен элементтеріне мәндер тағайындалады. Мысалы, `WIDTH, HEIGHT = bext.size()`.

Шарт операторлары (if-elif-else):

- Шарттарға байланысты орындалатын код блоктарын анықтаңыз. Бағдарлама терминалдың шеттерімен логотиптердің соқтығысуын тексеру үшін қолданылады.

Цикл операторы (for):

- Элементтер тізбегі бойынша Итерация үшін қолданылады. Бұл жағдайда логотиптерді жасау және оларды циклде өңдеу үшін.

Шексіз цикл (while True):

- Пайдаланушы үзгенге дейін Бағдарламаның негізгі логикасын орындайтын шексіз цикл жасайды.

Sys операторы.stdout.flush():

- Шығыс буферін қалпына келтіру үшін қолданылады. Бұл тұрғыда терминалға дұрыс шығуды қамтамасыз ету үшін қолданылады.

Терминалды шығару операторлары (print):

- Терминалға ақпаратты шығаруға жауапты. Мысалы, басып шығару (`'Corner bounces:', cornerBounces, end=""`).

Арифметика операторлары (+, -, *):

- Тағайындалған айнымалылардың мәндерін өзгерту үшін қолданылады. Мысалы, `cornerBounces += 1`.

Sys операторы.exit():

- Бағдарламаның орындалуын тоқтатады. Бұл жағдайда қажетті кітапхана болмаған жағдайда бағдарламаны аяқтау үшін қолданылады.

Уақыт операторы.sleep():

- Анимация жылдамдығына әсер ететін цикл итерациялары арасында кідіріс жасау үшін қолданылады.

Бұл операторлар терминалда секіретін DVD логотипінің анимациясын жасайтын бағдарламаның логикасын бірлесіп қамтамасыз етеді.

Қорытынды

Бұл бағдарлама қарапайым, бірақ қызықты бағдарламаны жүзеге асырады: терминалдағы DVD логотипінің анимациясы. Бағдарлама терминалмен, түстермен және уақытпен жұмыс істеу үшін Python бағдарламалау тілін және бірнеше кітапханаларды пайдаланады.

Ол `beht` кітапханасының бар-жоғын тексеруден басталады және егер ол жоқ болса, оны аяқтайды. Содан кейін терминал өлшемдерін, логотиптер санын, түстерді және басқа параметрлерді көрсететін тұрақтылар инициализацияланады.

Цикл арқылы түс, координаттар және қозғалыс бағыты сияқты кездейсоқ сипаттамалары бар логотиптер жасалады. Әрі қарай, шексіз циклде бағдарлама әр логотиптің қозғалысын, олардың терминалдың шеттері мен бұрыштарынан секірулерін өңдейді, секірулер кезінде түстерді өзгертеді, содан кейін логотиптерді жаңа позицияларға жылжытады және тартады.

Операторлар шарттар, циклдар және арифметикалық операторлар логотиптердің қозғалысын, серпілуін және дисплейін басқару үшін қолданылады. Бағдарлама сонымен қатар үзілісті өңдеуді қамтиды, осылайша пайдаланушы `Ctrl+C` пернелерін басу арқылы бағдарламаны аяқтай алады.

Бұл бағдарлама терминалдағы қарапайым графикалық анимацияның мысалы ретінде қызмет етеді және оны Python бағдарламалау негіздерін үйрену үшін оқу мақсатында пайдалануға болады.

Қолданылған әдебиеттер тізімі

- 1 Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
- 2 Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
- 3 Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
- 4 Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
- 5 Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
- 6 Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – 768 с.
- 7 Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. – 592 с.
- 8 Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O'Reilly»).
- 9 Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – 480 с.: – (Серия «Бестселлеры O'Reilly»).

Қосымша А

```
import sys, random, time
```

```
try:
```

```
    import bext
```

```
except ImportError:
```

```
    print("This program requires the bext module...")
```

```
    sys.exit()
```

```
WIDTH, HEIGHT = bext.size()
```

```
WIDTH -= 1
```

```
NUMBER_OF_LOGOS = 5
```

```
PAUSE_AMOUNT = 0.2
```

```
COLORS = ['red', 'green', 'yellow', 'blue', 'magenta', 'cyan', 'white']
```

```
UP_RIGHT = 'ur'
```

```
UP_LEFT = 'ul'
```

```
DOWN_RIGHT = 'dr'
```

```
DOWN_LEFT = 'dl'
```

```
DIRECTIONS = (UP_RIGHT, UP_LEFT, DOWN_RIGHT, DOWN_LEFT)
```

```
COLOR = 'color'
```

```
X = 'x'
```

```
Y = 'y'
```

```
DIR = 'direction'
```

```
def main():
```

```
    bext.clear()
```

```
    logos = []
```

```
    for i in range(NUMBER_OF_LOGOS):
```

```
        logos.append({COLOR: random.choice(COLORS),
```

```
                      X: random.randint(1, WIDTH - 4),
```

```
                      Y: random.randint(1, HEIGHT - 4),
```

```
                      DIR: random.choice(DIRECTIONS)})
```

```
    if logos[-1][X] % 2 == 1:
```

```
        logos[-1][X] -= 1
```

```
    cornerBounces = 0
```

```
    while True:
```

```
        for logo in logos:
```

```
            bext.goto(logo[X], logo[Y])
```

```

print(' ', end=")

originalDirection = logo[DIR]

if logo[X] == 0 and logo[Y] == 0:
    logo[DIR] = DOWN_RIGHT
    cornerBounces += 1
elif logo[X] == 0 and logo[Y] == HEIGHT - 1:
    logo[DIR] = UP_RIGHT
    cornerBounces += 1
elif logo[X] == WIDTH - 3 and logo[Y] == 0:
    logo[DIR] = DOWN_LEFT
    cornerBounces += 1
elif logo[X] == WIDTH - 3 and logo[Y] == HEIGHT - 1:
    logo[DIR] = UP_LEFT
    cornerBounces += 1
elif logo[X] == 0 and logo[DIR] == UP_LEFT:
    logo[DIR] = UP_RIGHT
elif logo[X] == 0 and logo[DIR] == DOWN_LEFT:
    logo[DIR] = DOWN_RIGHT
elif logo[X] == WIDTH - 3 and logo[DIR] == UP_RIGHT:
    logo[DIR] = UP_LEFT
elif logo[X] == WIDTH - 3 and logo[DIR] == DOWN_RIGHT:
    logo[DIR] = DOWN_LEFT
elif logo[Y] == 0 and logo[DIR] == UP_LEFT:
    logo[DIR] = DOWN_LEFT
elif logo[Y] == 0 and logo[DIR] == UP_RIGHT:
    logo[DIR] = DOWN_RIGHT
elif logo[Y] == HEIGHT - 1 and logo[DIR] == DOWN_LEFT:
    logo[DIR] = UP_LEFT
elif logo[Y] == HEIGHT - 1 and logo[DIR] == DOWN_RIGHT:
    logo[DIR] = UP_RIGHT

if logo[DIR] != originalDirection:
    logo[COLOR] = random.choice(COLORS)

if logo[DIR] == UP_RIGHT:
    logo[X] += 2
    logo[Y] -= 1
elif logo[DIR] == UP_LEFT:
    logo[X] -= 2
    logo[Y] -= 1
elif logo[DIR] == DOWN_RIGHT:
    logo[X] += 2

```

```

        logo[Y] += 1
    elif logo[DIR] == DOWN_LEFT:
        logo[X] -= 2
        logo[Y] += 1

    bext.goto(5, 0)
    bext.fg('white')
    print('Corner bounces:', cornerBounces, end=")

    for logo in logos:
        bext.goto(logo[X], logo[Y])
        bext.fg(logo[COLOR])
        print('DVD', end=")

    bext.goto(0, 0)
    sys.stdout.flush()
    time.sleep(PAUSE_AMOUNT)

if __name__ == '__main__':
    try:
        main()
    except KeyboardInterrupt:
        print()
        print('Bouncing DVD Logo, by Al Sweigart')
        sys.exit()

```

Анықтама

Берілген жұмыс бойынша онлайн тексеріс

Кафедра:
Топ нөмері:
Жұмыс түрі:
Пән:

АЕЖ
ВТ-20-1
Курстық жоба
Функционалдық және логикалық бағдарламалау



Отчет предоставлен сервисом
«Антиплагиат» - <http://users.antiplagiat.ru>



Отчет о проверке

Автор: Серікжан Бауыржан

Проверяющий:

Название документа: ФЛПО ВТ-20-1 КР Серікжан Бауыржан

РЕЗУЛЬТАТЫ ПРОВЕРКИ

Тариф: FREE

Совпадения: Не менее 0,51%
Оригинальность: Не более 99,49%

Цитирования: Недоступно для FREE*
Самоцитирование: Недоступно для FREE*

*«Совпадения», «Цитирования», «Самоцитирование», «Оригинальность» являются отдельными показателями, отображаются в процентах и в сумме дают 100%, что соответствует полному тексту проверяемого документа.

*Результаты проверки на тарифе FREE являются неполными и ограниченными по сравнению с тарифом FULL и корпоративной версией, так как проверка идет по источникам, добавленным до 15 ноября 2021 года, с использованием урезанных возможностей системы

- Совпадения** — фрагменты проверяемого текста, полностью или частично сходные с найденными источниками, за исключением фрагментов, которые система отнесла к цитированию или самоцитированию. Показатель «Совпадения» — это доля фрагментов проверяемого текста, отнесенных к совпадениям, в общем объеме текста.
- Самоцитирование** — фрагменты проверяемого текста, совпадающие или почти совпадающие с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа. Показатель «Самоцитирование» — это доля фрагментов текста, отнесенных к самоцитированию, в общем объеме текста.
- Цитирование** — фрагменты проверяемого текста, которые не являются авторскими, но которые система отнесла к корректно оформленным. К цитированиям относятся также шаблонные фразы, библиография, фрагменты текста, найденные модулем поиска «СПС Гарант: нормативно-правовая документация». Показатель «Цитирование» — это доля фрагментов проверяемого текста, отнесенных к цитированию, в общем объеме текста.
- Текстовое пересечение** — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
- Источники** — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
- Оригинальный текст** — фрагменты проверяемого текста, не обнаруженные ни в одном источнике и не отмеченные ни одним из модулей поиска. Показатель «Оригинальность» — это доля фрагментов проверяемого текста, отнесенных к оригинальному тексту, в общем объеме текста.

Обращаем Ваше внимание, что система находит текстовые совпадения проверяемого документа с проиндексированными в системе источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности совпадений или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

Номер документа: 2	Количество страниц: 9
Тип документа: Курсовая работа	Символов в тексте: 40951
Дата проверки: 09.12.2023 01:23:54	Слов в тексте: 4737
Дата корректировки: 09.12.2023 01:33:05	Число предложений: 392

Комментарий: не указано

ПАРАМЕТРЫ ПРОВЕРКИ

Выполнена проверка с учетом редактирования: Нет
Выполнено распознавание текста (OCR): Нет
Выполнена проверка с учетом структуры: Нет

Модули поиска: Интернет Free

ИСТОЧНИКИ

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	0.51%	0.51%	Білім беру бағдарламасы «Компьютерлік инженерия» http://kazatu.kz	20 Дек 2019	Интернет Free

Жоба жетекші: Искаков М.Б
Студент: Зекен Алихан