

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук,  
управління та адміністрування  
Кафедра Інформаційних технологій

**БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему: Розробка модульного кросплатформного програвача звукових  
файлів з візуалізацією

Виконав студент 4 курсу групи К-25  
Спеціальність 122 Комп'ютерні науки  
Гажук Владислав Ігорович

Керівник ст.викл.  
Рольщиков Вадим Борисович

Консультант к.геогр.н., доцент  
Кузніченко Світлана Дмитрівна

Рецензент к.геогр.н., доцент  
Лужбін Анатолій Михайлович

## ЗМІСТ

Скорочення та умовні позначки .....	6
Вступ.....	7
1 Аналіз Предметної Області .....	9
Функціональності медіаплеєрів .....	9
Робота та використання 3D-відеоплеєрів .....	10
Програмне забезпечення.....	12
2 Вибір Програмних Засобів.....	14
Вибір мови програмування .....	14
Особливості та філософія мови Python.....	15
Синтаксис і семантика в мові програмування Python.....	17
Бібліотеки мови Python .....	19
Середовища розробки.....	21
Реалізації створення програмного забезпечення в мови Python .....	21
Вибір мови для обробки зовнішнього вигляду .....	26
Синтаксис мови CSS.....	27
Одиниці виміру довжини в CSS .....	28
Використання мови CSS.....	28
Спадкування в мови CSS.....	30
Варіації версії мови CSS .....	31
Характеристика мови CSS версії 1 .....	31
Характеристика мови CSS версії 2.....	32
Характеристика мови CSS версії 2.1 .....	32
Характеристика мови CSS версії 3 .....	33
Характеристика мови CSS версії 4.....	33
Підтримка веб-браузерів.....	34
Можливості мови CSS .....	35
Середовище розробки зовнішнього вигляду QT .....	36

	5
Концепції програми Qt .....	38
Видання програми Qt.....	40
Підтримувані платформи Qt .....	41
Ліцензування програмного забезпечення.....	42
Інструменти Qt .....	42
3 Реалізація Програмної Частини .....	44
Створення аудіо програвача .....	44
Створення списку аудіо файлів.....	46
Розробка зовнішнього вигляду програвача .....	49
Перетворення Python в автономні виконувані файли.....	52
Висновки .....	53
Перелік джерел посилання .....	54

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОДЕКУ – Одеський Державний Екологічний Університет

ПЗ – Програмне Забезпечення

Мр3– сжатый звуковий файл

Ogg – сжатый звуковий файл

Wav – Waveform Audio File

Python– мова програмування

Css – мова розмітки

Windows – операційна система

Linux – операційна система

MacOS – операційна система

IDE – integrated development environment(Інтегрована середовище розробки)

## ВСТУП

Що потрібно від хорошого програмного програвача? В першу чергу – всеїдність. Софт повинен розуміти всі актуальні для меломанів формати аудіофайлів. Не менш важливо зручність використання. Крім того, плеєр повинен коректно виводити аудіопотоки на апаратний конвертор і делікатно проводити обробку звуку при необхідності.

Найбільш популярним серед меломанів є кодування зі стиском без втрат, так звані lossless-файли. Спеціальні алгоритми дозволяють зберегти весь обсяг звукової інформації, скоротивши розмір файлу, в середньому, удвічі. Найчастіше записи lossless мають ще і високу роздільну здатність (аж до 24 біт / 192 кГц), тому сучасний програмний плеєр повинен вміти працювати в повній мірі і з таким контентом. А також з різними образами – файловими копіями оптичних носіїв Audio CD або SACD. Вони можуть зберігатися у вигляді архівів (наприклад, ISO), або у вигляді неархівованих тандемів CUE + APE / FLAC. На відміну від одиночних треків, тандеми влаштовані інакше: весь альбом або збірник зберігається одним аудіофайлом, а точки початку кожного треку і назви доріжок прописані в списку CUE, своєрідною навігаційній карті для плеєра. Відповідно, якщо альбом задуманий авторами як цільне звукове полотно з плавно перетікають одна в одну композиціями, то ви зможете їх слухати без стрибків і пауз, при цьому збережете можливість переміщатися між треками. Якщо ж подібний альбом закодований у вигляді окремих доріжок, то плеєр повинен підтримувати функцію Gapless – автоматичне видалення пауз по краях записи для непомітного переходу від кінця одного файлу в початок іншого.

Меломани пред'являють до сучасних плеєрів ряд додаткових вимог. Одне з них – підтримка студійного стандарту ASIO. У багатьох внутрішніх аудіокарта і зовнішніх USB-ЦАП є відповідний режим роботи двайвера, він дозволяє направляти звукові потоки з плеєра прямо в пристрій, мінаючи програмні засоби обробки та маршрутизації операційної системи. Що це дає

на практиці? Передачу даних з побітової точністю. Будь-які маніпуляції з аудіопотоків відбуваються тільки з відома користувача! Найбільш популярною з свідомо скоєних оптимізацій є еквалізація. Хай вибачать нас аудіофіли, але в недорогих системах, встановлених в звичайних житлових приміщеннях, частотна корекція несе більше користі, ніж шкоди. Крім того, еквалайзер еквалайзера ворожнечу. У студіях звукозапису такі прилади (точніше, все частіше їх програмні аналоги) використовуються повсюдно, але це не заважає отримувати на виході шикарне якість тембрів і голографічну звукову сцену. В сучасних меломанським програмних плеєрах для комп'ютера еквалайзери, як правило, дуже гідні. Вони виконують обробку звуку в 32-бітному форматі, тому «супутні втрати» у вигляді фазових спотворень якщо і чути, то рідко і на дуже дорогій техніці. При цьому еквалайзер здорово допомагає вирівняти АЧХ колонок або навушників, компенсуючи «провали» і усунувши «горби». Зрозуміло, сам користувач повинен чітко розуміти що і як треба «підкрутити», щоб прослуховування доставляло ще більше задоволення. Наприклад, для недопущення клипінга і компресії (слідства спрацьовування автоматичного захисту від перевантаження) за нульовий рівень при налаштуванні еквалайзера краще не виходити: якщо потрібно підняти якийсь діапазон, припустимо, на 4 дБ, то потрібно не обрана ділянка піднімати, а решта збавляти.

Сьогодні для комп'ютерів на базі Windows пропонується безліч програмних плеєрів, більшість з яких повністю безкоштовні. Однак значна частина цих рішень не в повній мірі відповідає вимогам сучасного меломана, який використовує ПК в якості основного засобу прослуховування музики. Ми відібрали для вас три найцікавіших, на наш погляд, програвача, які можемо рекомендувати з повною упевненістю.

Пояснювальна записка містить 55 сторінок, 14 рисунків, 3 таблиці і 14 джерел посилання.

## 1 Аналіз предметної області

Програваач мультимедіа (медіа-плеєр, англ. Media player) –це тип прикладного програмного забезпечення для відтворення мультимедійних комп'ютерних файлів, таких як аудіо і відео файли. Медіаплеєри зазвичай відображають стандартні значки управління мультимедіа, відомі з фізичних пристроїв, таких як магнітофони і CD-плеєри, такі як кнопки play (▶), pause (⏸), fastforward, backforward і stop (■). Крім того, вони зазвичай мають індикатори виконання (або "панелі відтворення"), щоб знайти поточне положення в тривалості мультимедійного файлу.

Основні операційні системи мають принаймні один медіаплеєр за замовчуванням. Наприклад, Windows поставляється з Windows Media Player, Microsoft Movies & TV і Groove Music, а macOS QuickTime Player і Music . Дистрибутиви Linux поставляються з різними медіа-плеєрами, такими як SMPlayer , Amarok , Audacious , Banshee , MPlayer , mpv , Rhythmbox , Totem , VLC media player і xine . Android поставляється з Google Play Music для аудіо і Google Photos для відео.

### Функціональності медіаплеєрів

Різні медіа-плеєри мають різні цілі та набори функцій. Відеоплеєри-це група медіаплеєрів, які мають свої функції, спрямовані більше на відтворення цифрового відео. Наприклад, Windows DVD-плеєр виключно відтворює DVD-відео диски і нічого більше. Media Player Classic може відтворювати окремі аудіо і відео файли, але багато хто з його функцій, таких як корекція кольору, різкість зображення, масштабування, набір гарячих клавіш, підтримка DVB, корисні тільки для відеоматеріалів, таких як фільми та мультфільми. Аудіоплеєри, з іншого боку, спеціалізуються на цифровому аудіо. Наприклад, AIMP виключно відтворює аудіоформати. MediaMonkey може відтворювати як аудіо, так і відео формати, але багато хто з його

функцій, включаючи медіатеку, ліричний виявлення, візуалізацію музики, онлайн-радіо, індексування аудіокниг і редагування тегів, орієнтовані на споживання аудіоматеріалу; перегляд відеофайлів на ньому може бути проблемним подвигом. Універсальні медіаплеєри також існують. Наприклад, Програмач Windows Media має ексклюзивні опції для аудіо та відеоматеріалів, хоча він не може відповідати набору функцій Media Player Classic і MediaMonkey йогу функціональність плеєра на рис 1.

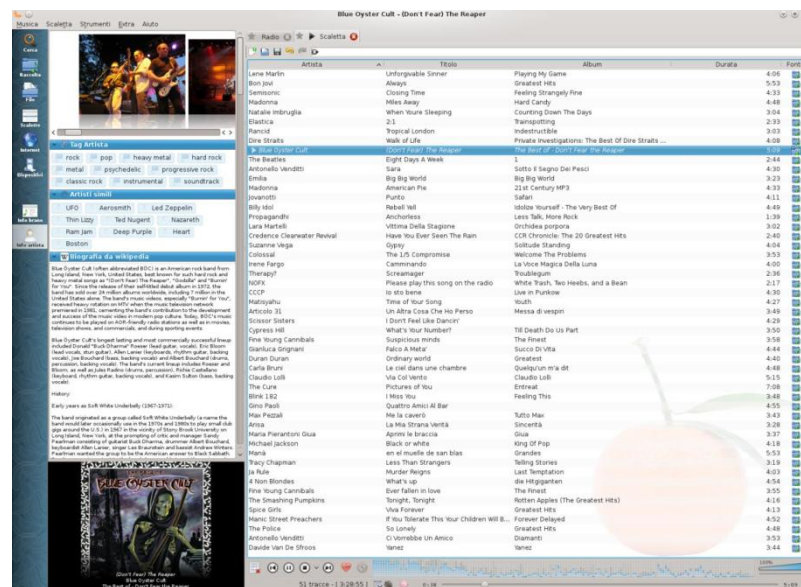


Рисунок 1 – зовняшній вигляд мдіаплеєра

## Робота та використання 3D-відеоплеєрів

3D-відеоплеєри використовуються для відтворення 2D-відео в форматі 3D. Високоякісна тривимірна відео презентація потрібно, щоб кожен кадр кінофільму був вбудований з інформацією про глибину об'єктів, присутніх на сцені. Цей процес включає в себе зйомку відео з допомогою спеціального обладнання з двох різних точок зору або моделювання та візуалізації кожного кадру у вигляді набору об'єктів, що складаються з 3D вершин і текстур, так само, як і в будь-якій сучасній відеогрі, для досягнення спеціальних ефектів. Виснажливий і дорогий, цей метод використовується лише у невеликої част-



ки фільмів, вироблених в усьому світі, в той час як більшість фільмів залишаються у вигляді традиційних 2D-зображень. Однак в іншому випадку можна надати двовимірної картині видимість глибини. Використовуючи метод, відомий як анагліфическая обробка "плоского" зображення, його можна перетворити так, щоб створити ілюзію глибини при перегляді через анагліфні окуляри. Зображення, розглянуте через анагліфічних окуляри, мабуть, має як доповідачі, так і глибоко вбудовані об'єкти, за рахунок дещо спотворених квітів. Сам метод досить старий починаючи з середини 19-го століття, але тільки з останніми досягненнями в області комп'ютерних технологій стало можливим застосовувати цей вид перетворення до ряду кадрів в кінофільмі досить швидко або навіть в реальному часі, тобто при відтворенні відео. Існує декілька реалізацій у вигляді 3D-відеоплеєрів, які візуалізують звичайне 2D-відео в форматі anaglyph 3D, а також у вигляді 3D-відео конвертерів, які перетворюють відео в стереоскопічний анагліф і перекодирують його для відтворення з звичайними програмними або апаратними відеоплеєрами зображен на рис. 2.

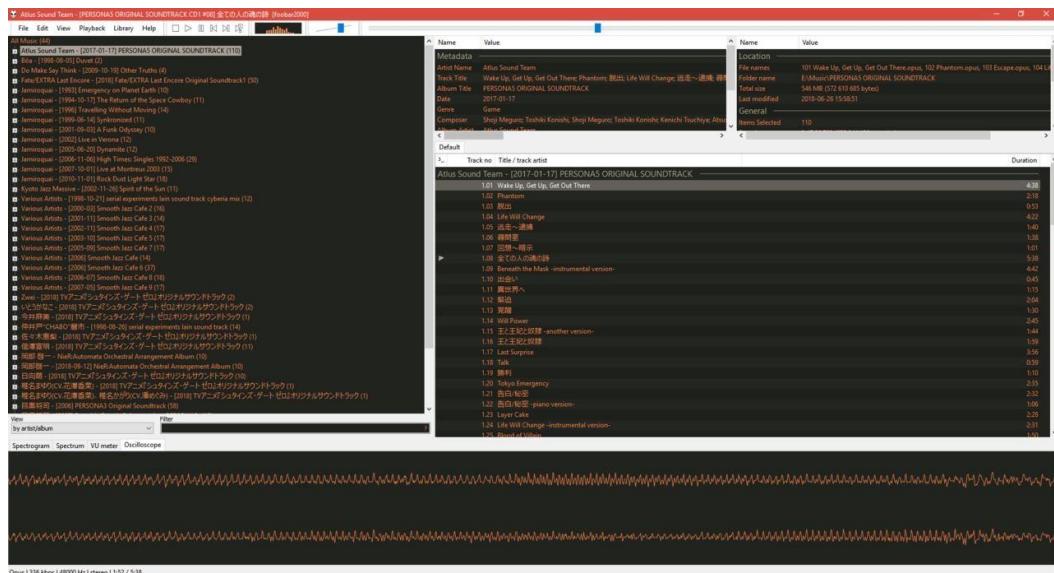


Рисунок 2 – зовнішній вигляд апаратний відеоплеєр AST

## Програмне забезпечення

Багато мультимедіа програвачів використовують бібліотеки, які призначені для систематизації й каталогізації музики й відео по категоріях, таким як жанр, рік запису, оцінка й ін. Гарним прикладом мультимедіа програвачів, які використовують бібліотеки є Winamp, Windows Media Player, iTunes, RealPlayer і Amarok. Список деяких мультимедіа програвачів можна подивитися у таблиці 1

Таблиця 1 – Список мультимедійних програвачів

Програмні мультимедіа програвачі				
Назва	Аудіо	Відео	Операційна система	Ліцензія
1	2	3	4	5
Adobe Media Player	Так	Так	Windows, Mac OS X	власна
Aimp	Так	Ні	Windows	GPL
Amarok	Так	Ні	POSIX-сумісні	GPL
aTunes	Так	Ні	Windows, Linux, Solaris, Mac OS X, FreeBSD	GPL
Audacious	Так	Ні	POSIX-сумісні	GPL
Banshee	Так	Ні	POSIX-сумісні	MIT
Bearshare	Так	Ні	Windows	власна
Beep Media Player	Так	Ні	POSIX-сумісні	GPL
BS.Player	Так	Так	Windows	власна
cmus	Так	Ні	POSIX-сумісні	GPL
DeaDBeeF	Так	Ні	UNIX-подібні	GPL, LGPL
Foobar2000	Так	Ні	Windows	власна
GOM Player	Так	Так	Windows	власна
iMesh	Так	Ні	Mac, Windows	власна
iTunes	Так	Так	Mac, Windows	власна
jetAudio	Так	Так	Windows	власна,
Kaffeine Player	Так	Так	POSIX-сумісні	GPL

## Продовження таблиці 1

1	2	3	4	5
Light Alloy	Так	Так	Windows	власна
MediaMonkey	Так	Hi	Windows	власна
Media Player Classic	Так	Так	Windows	GPL
MPlayer	Так	Так	POSIX-сумісні, Mac OS X, Windows, AmigaOS, MorphOS	GPL
mpv	Так	Так	Linux/Unix, Windows, Mac OS X, BeOS, BSD	GPL
MusicMatch Jukebox	Так	Hi	Windows	власна
Napster	Так	Hi	Windows	власна
Narrowstep Player	Так	Так	BeI	власна
PowerDVD	Hi	Так	Windows	власна (CyberLink)
Quicktime	Так	Так	Mac OS X, Windows	власна (Apple)
RealPlayer	Так	Так	Linux, Windows, Mac OS X, Palm OS, Symbian OS	власна
Rhythmbox	Так	Hi	POSIX-сумісні	GNU LGPL
Snow Player	Так	Так	Windows	власна
Quod Libet	Так	Hi	POSIX-сумісні, Mac OS X, Windows, AmigaOS, MorphOS	GPL
Windows Media Player	Так	Так	Windows, Mac OS X	власна
XBMC (XBox Media Center)	Так	Так	Xbox game-console, Windows (partial port to Win32)	GPL /LGPL
xine	Так	Так	Linux, FreeBSD, Solaris, IRIX, Mac OS X	GPL
XMMS	Так	Hi	POSIX-сумісні	GPL
Zinf	Так	Hi	Linux, Windows	GPL
Zoom Player	Так	Так	Windows	власна

## 2 Вибір програмних засобів

### Вибір мови програмування

Для вибору мови програмування було обрано мову Python

Python – це інтерпретується в microsoft, високорівневий, універсальна мова програмування . Створена Guido van Rossum і вперше випущена в 1991 році, філософія дизайну Python підкреслює читаність коду з його помітним використанням значних прогалів. Його мовні конструкції і об'єктно-орієнтований підхід спрямовані на те, щоб допомогти програмістам писати ясний, логічний код для невеликих і великомасштабних проєктів.[1]<sup>1</sup>

Python динамічно типизується і збирається сміття. Він підтримує декілька парадигм програмування, включаючи структуроване (зокрема, процедурне), об'єктно-орієнтоване та функціональне програмування . Python часто описується як мову через його всеосяжної стандартної бібліотеки.

Логотип мови програмування Python зображено на рис. 3

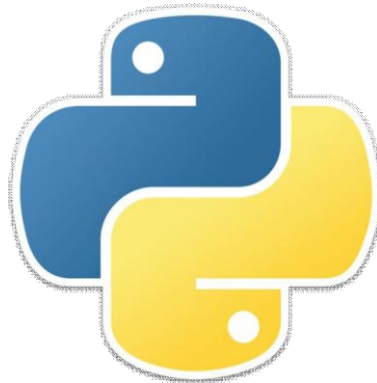


Рисунок 3 – Логотип Python

---

<sup>1</sup> [1] Кульман, Дэйв. "Книга на Python: начало Python, расширенный Python и Python упражнения" URL: [https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python\\_book\\_01.html](https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html)

Python був задуманий в кінці 1980-х років як спадкоємець мови ABC. Python 2.0, випущений в 2000 році, представив такі функції, як розуміння списку і система збору сміття, здатна збирати посилальні цикли. Python 3.0, випущений в 2008 році, був серйозною ревізією мови, який не повністю назад сумісний, і велика частина коду Python 2 не виконується незмінним на Python 3.

Мова Python 2 була офіційно припинено у 2020 році (вперше заплановано на 2015 рік), і "Python 2.7.18 є останнім релізом Python 2.7 і, отже, останнім релізом Python 2." [2]<sup>1</sup> більше ніяких виправлень безпеки або інших поліпшень для нього не буде випущено. З закінченням терміну служби Python 2, тільки Python 3.5. і більш пізні версії.

перекладачі доступні для багатьох операційних систем. Глобальне співтовариство програмістів розробляє і підтримує CPython, відкритий вихідний . Некомерційна організація, Python Software Foundation, керує і спрямовує ресурси для розробки Python і CPython.

### **Особливості та філософія мови Python**

Об'єктно-орієнтоване програмування та структуроване Програмування повністю підтримуються, і багато з його функцій підтримують функціональне програмування і аспектно-орієнтоване програмування (в тому числі з допомогою метапрограмування і метаоб'єктів (магічних методів)). багато інших парадигми підтримуються з допомогою розширень, включаючи проектування за контрактом і логічне програмування.

Python використовує динамічне типування і комбінацію підрахунку послань і збирача сміття з виявленням циклів для управління пам'яттю. Він також має динамічний дозвіл імен, яке пов'язує імена методів і змінних під час виконання програми.

---

<sup>1</sup> [2] Peterson, Benjamin (20 Апреля 2020). "Python Insider: Python 2.7.18, последний релиз Python 2" URL: <https://pythoninsider.blogspot.com/2020/04/python-2718-last-release-of-python-2.html>

Дизайн Python пропонує деяку підтримку для функціонального програмування в традиції. Він має `filtermapі reduce` функції; список поглядів, словники, набори та генератори виразів.[3]<sup>1)</sup> стандартна бібліотека має два модулі (`itertools` і `functools`), які реалізують функціональні інструменти, запозичені з Haskell і Standard ML.

Основна філософія мови коротко викладена в документі *The Zen of Python* (PEP 20)[4]<sup>2)</sup>, який включає в себе такі елементи, як:

- красива-це краще, ніж потворна;
- явне краще, ніж неявне;
- просте краще, ніж складне;
- складне краще, ніж складне;
- читаємість має значення.

Замість того, щоб мати всю свою функціональність, вбудовану в його ядро, Python був розроблений, щоб бути дуже розширюваним. Ця компактна модульність зробила його особливо популярним в якості засобу додавання програмованих інтерфейсів до існуючих додатків. Бачення Ван Россумом невеликого базової мови з великою стандартною бібліотекою і легко розширюваним інтерпретатором виникало з його розчарування в ABC, який підтримував протилежний підхід.

Python прагне до більш простої, менш захащеному синтаксису і граматики, надаючи розробникам вибір методології їх кодування. На відміну від девізу Perl "є більш ніж один спосіб зробити це", Python охоплює "повинен бути один—і переважно тільки один—очевидний спосіб зробити це" філософія дизайну. Алекс Мартелли, співробітник Python Software Foundation і автор книги Python, пише, що "описувати щось як "розумне" не вважається компліментом в культурі Python."

Розробники Python прагнуть уникнути передчасної оптимізації і відкидають патчі до некритичним частинам еталонної реалізації, які пропо-

---

<sup>1</sup> [3] Hettinger, Raymond (30 января 2002 года). "PEP 289-генератор выражений" URL: <https://www.python.org/dev/peps/pep-0289/>

<sup>2</sup> [4] Peters, Tim (19 августа 2004 года). "PEP 20-Дзен питона" URL: <https://www.python.org/dev/peps/pep-0020/>

нували б незначне збільшення швидкості за рахунок ясності. коли важлива швидкість, програміст Python може переміщати критично важливі за часом функції у модулі розширень, написані на таких мовах, як C, або використовувати Cython, компілятор just-in-time. також доступна функція, яка переводить скрипт на Python мова C і виконує прямі виклики API рівня C інтерпретатор Python.

Загальним неологізмом в співтоваристві Python є *pythonic*, який може мати широкий спектр значень, пов'язаних зі стилем програми. Сказати, що код є питонським, означає сказати, що він добре використовує ідіоми Python, що це природно або показує швидкість в мові, що він відповідає мінімалістській філософії Python і акценту на читаність. Навпаки, код, який важко зрозуміти або читається як груба транскрипція з іншої мови програмування, називається *unpythonic*.

Користувачів і шанувальників Python, особливо тих, хто вважається знають або досвідченими, часто називають Питоністами.

## Синтаксис і семантика в мові програмування Python

Python має бути легко читаним мовою. Його форматування візуально лаконічно, і він часто використовує англійські ключові слова, де інші мови використовують пунктуацію. На відміну від багатьох інших мов, він не використовує фігурні дужки для розмежування блоків, а точки з комою після операторів є необов'язковими. Він має менше синтаксичних винятків і особливих випадків, ніж C або Pascal.[5]<sup>1</sup>

Python використовує відступ, а не фігурні дужки або ключові слова, щоб розмежувати блоки. Збільшення відступу відбувається після певних тверджень; зменшення відступу означає кінець поточного блоку. таким чином, візуальна структура програми точно представляє семантичну структуру програми. Цю особливість іноді називають правилом *off-side*, яке поділяють

---

<sup>1</sup> [5] "Є гарною мовою Python для початківців програмістів?". URL: <https://docs.python.org/3/faq/general.html#is-python-a-good-language-for-beginning-programmers>

деякі інші мови, але в більшості мов відступ не має ніякого семантичного значення.

Ведення типів ієрархії зображен на рис. 4

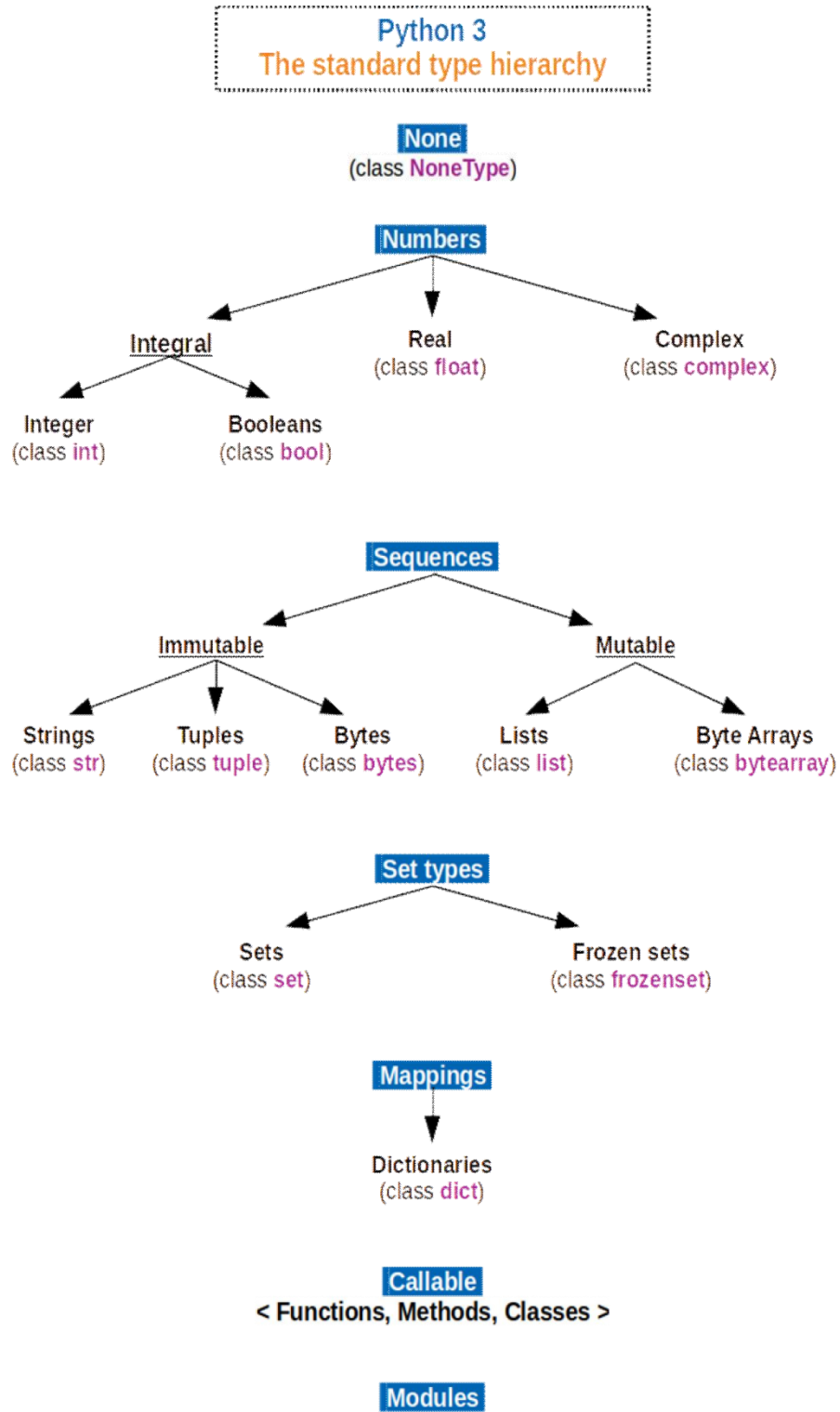


Рисунок 4 – Стандартна ієрархія типів в Python 3



Python використовує duck typing і має типізовані об'єкти, але нетипізовані імена змінних. Обмеження типу не перевіряються під час компіляції ; швидше за все, операції над об'єктом можуть завершитися помилкою, що означає, що даний об'єкт не є підходящим типом. Незважаючи на динамічну типізацію, Python є строго типізованим, забороняючи операції, які не є добре визначеними (наприклад, додавання числа в рядок), а не мовчки намагаються зрозуміти їх.

Python дозволяє програмістам визначати свої власні типи з допомогою класів, які найчастіше використовуються для об'єктно-орієнтованого програмування. Нові екземпляри класів створюються шляхом виклику класу (наприклад, `SpamClass()` or `EggsClass()`), і класи є екземплярами метакласу `type` (сам примірник себе), дозволяючи метапрограмування і відображення.

До версії 3.0 в Python було два типи класів: старий стиль і новий стиль. синтаксис обох стилів однаковий, різниця полягає в тому, успадковується клас об'єктнопрямо або побічно (всі класи нового стилю успадковують об'єкти є екземплярами `type`). У версіях Python 2 починаючи з Python 2.2 і далі можуть використовуватися обидва типи класів. Класи старого стилю були виключені в Python 3.0.

Довгостроковий план полягає у підтримці поступового введення і від Python 3.5, синтаксис мови дозволяє вказати статичні типи, але вони не перевіряються реалізації за замовчуванням, CPython. Експериментальна додаткова статична перевірка типів з ім'ям `typing` підтримує перевірку типів під час компіляції.

## **Бібліотеки мови Python**

Велика бібліотека Python, зазвичай згадується як одна з її найсильніших сторін, надає інструменти, що підходять для багатьох завдань. Для додатків з виходом в інтернет підтримуються багато стандартні формати і протоколи, такі як MIME і HTTP. Вона включає в себе модулі для створення графічних користувацьких інтерфейсів , підключення до реляційних баз

даних, генерації псевдовипадкових чисел, арифметики з десятковими дробами довільної точності, маніпулювання регулярними виразами і модульного тестування.

Деякі частини стандартної бібліотеки охоплені специфікаціями (наприклад, реалізація інтерфейсу шлюзу веб-сервера (WSGI) `wsgiref` слід за PEP 333[6]<sup>1</sup>), але більшість модулів-ні. Вони визначаються їх кодом, внутрішньою документацією та комплектами тестів. Однак, оскільки більша частина стандартної бібліотеки є кросплатформним кодом Python, тільки кілька модулів потребують зміни або переписуванні для різних реалізацій.

Станом на листопад 2019 року, Python Package Index (PyPI), офіційний репозиторій для стороннього програмного забезпечення Python, містить понад 200 пакетів із широким спектром функціональних можливостей, включаючи:

- автоматизація;
- аналіз даних;
- база даних;
- документація;
- графічний користувальницький інтерфейс;
- обробка зображень;
- машинне навчання;
- мобільний додаток;
- мультимедіа;
- мережевий;
- наукові обчислення;
- системне адміністрування;
- тестові платформи;
- обробка тексту;
- веб-фреймворк;
- зішкріб полотна.

---

<sup>1</sup> [6] Eby, Phillip J. (7 Декабря 2003 Года). "PEP 333-Python Web Server Gateway Interface v1.0" URL: <https://www.python.org/dev/peps/pep-0333/>

## Середовища розробки

Більшість реалізацій Python (включаючи CPython) включають цикл read-eval-print loop (REPL), що дозволяє їм функціонувати в якості інтерпретатора командного рядка, для якого користувач вводить оператори послідовно і відразу отримує результати.

Інші оболонки, включаючи IDLE і IPython, додають додаткові можливості, такі як покращене автоматичне завершення, збереження стану сеансу та підсвічування синтаксису.

Крім стандартних настільних інтегрованих середовищ розробки, існують веб-браузерні IDE; SageMath (призначені для розробки наукових і математичних програм Python); PythonAnywhere, браузерна середовище IDE і середовище хостингу; і Canopy IDE, комерційна Python IDE, підкреслює наукові обчислення.

## Реалізації створення програмного забезпечення в мови Python

Еталонна реалізація – CPython є еталонною реалізацією Python. Він написаний на мові C, що відповідає стандарту з кількома обраними функціями. Він компілює програми Python у проміжний байт<sup>-код</sup>, який потім виконується його віртуальною машиною. CPython поширюється з великою стандартною бібліотекою, написаною на суміші C і власного Python. Він доступний для багатьох платформ, включаючи Windows (Vista і більш пізні версії; підтримувані Windows XP і старше, з тепер не підтримуваних Python 2.7) і більшість сучасних Unix-подібних систем. Переносимість платформи була одним із його найперших пріоритетів, в Python 1 і 2 часові рамки, навіть підтримуючи віртуальні машини OS / 2, в той час як з тих пір підтримка була відкинута для багатьох платформ.

Інші реалізації. – PyPy-це швидкий, сумісний інтерпретатор Python 2.7 і 3.6. його just-in-time компілятор приносить значне поліпшення швидкості в

порівнянні з CPython, але кілька бібліотек, написаних на C, не можуть бути використані з ним.

Stackless Python є значною виделкою CPython, яка реалізує; він не використовує стек пам'яті C, що дозволяє масово паралельні програми. PyPy також має stackless версію.

MicroPython і CircuitPython-це Python 3 варіанти, оптимізовані для мікроконтролерів. Це включає в себе Lego Mindstorms EV3 .

Непідтримувані реалізації – Інші компілятори Python just-in-time були розроблені, але тепер не підтримуються:

Google почав проект під назвою Unladen Swallow в 2009 році, з метою прискорення інтерпретатора Python у п'ять разів з допомогою LLVM, а також поліпшення його многопоточної здатність масштабуватися до тисячі ядер, в той час як звичайні реалізації страждають від глобальної блокування інтерпретатора .

Psyco був як раз вчасно спеціалізований компілятор, який інтегрується з CPython і перетворює байт-код в машинний код під час виконання. Створений код є спеціалізованим для певних типів даних і працює швидше, ніж стандартний код Python.

У 2005 році Nokia випустила інтерпретатор Python для мобільних телефонів з ім'ям PyS60. Він включає в себе безліч модулів з реалізацій CPython і деякі додаткові модулі для інтеграції з операційною системою. Проект був оновлений, щоб працювати на всіх варіантах платформи S60, і доступні кілька сторонніх модулів. Nokia N900 також підтримує Python з бібліотеками віджетів, що дозволяє писати і запускати програми на цільовому пристрої.

Існує кілька компіляторів для об'єктних мов високого рівня, з необмеженим Python, обмеженою підмножиною Python або язиком, схожим на Python у якості вихідного мови:

- Cython компілює Python в C і c++;
- Google's Grumpy компілює Python, щоб йти;

- IronPython слід аналогічного підходу для запуску програм на Python.NET Common Language Runtime;
- Jython дозволяє використовувати бібліотеку класів Java програми з Python;
- MyHDL компілює Python в VHDL;
- Nuitka компілює Python у C++;
- Numba використовує LLVM для компіляції Python у машинний код;
- Pyjs компілює Python в JavaScript ;
- Pyrex (останній реліз в 2010 році) і Shed Skin (останній реліз в 2013 році) компілюються на C та C++ відповідно;
- Pythran компілює Python у C++;
- RPython може бути скомпільований в C, і використовується для побудови інтерпретатора PyPy Python.

Порівняння продуктивності різних реалізацій Python на нечислової (комбінаторної) робочій навантаженні було представлено на EuroSciPy '13. продуктивність Python у порівнянні з іншими мовами програмування також була оцінена за допомогою комп'ютерної тестової гри мови .

Починаючи з 2003 року, Python послідовно входить у десятку найбільш популярних мов програмування в індексі спільноти програмування ТЮВЕ, де станом на лютий 2020 він є третім за популярністю мовою (після Java і C). він був обраний мовою програмування року в 2007, 2010 і 2018 роках [7]<sup>1</sup>.

Емпіричне дослідження показало, що скриптові мови, такі як Python, більш продуктивні, ніж звичайні мови, такі як C і Java, для задач програмування, пов'язаних з маніпуляцією рядками і пошуком у словнику, і визначило, що споживання пам'яті часто було "краще, ніж Java, і не набагато гірше, ніж C або C++".

---

<sup>1</sup> [7] Prechelt, Lutz (14 Марта 2000 Года). Емпіричне порівняння C, C++, Java, Perl, Python, Rexx и Tcl URL: [http://page.mi.fu-berlin.de/prechelt/Biblio/jccpprt\\_computer2000.pdf](http://page.mi.fu-berlin.de/prechelt/Biblio/jccpprt_computer2000.pdf)

Великі організації, які використовують Python, включають Wikipedia, Google, Yahoo! Instagram Facebook , а також деякі більш дрібні організації, такі як сайт соціальної мережі новин Reddit, повністю написаний на Python.

Python може служити мовою сценаріїв для веб-додатків, наприклад, через `mod_wsgi` для веб-сервера Apache . з допомогою інтерфейсу шлюзу веб-сервера був розроблений стандартний API для полегшення цих додатків. Веб-фреймворки , такі як Django , Pylons , Pyramid , TurboGears , web2py , Tornado , Flask, Bottle і Zope, підтримують розробників у проектуванні та обслуговуванні складних додатків. Pyjs і IronPython можна використовувати для розробки клієнтської частини Ajax-додатків. SQLAlchemy можна використовувати як до реляційної бази даних. Кручений є основою для програмування зв'язку між комп'ютерами, і використовується, наприклад, Dropbox .

Бібліотеки, такі як NumPy, SciPy і Matplotlib дозволяють ефективно використовувати Python у наукових обчисленнях, зі спеціалізованими бібліотеками, такими як Biopython і Astropy, що забезпечують домен-специфічну функціональність. SageMath-це математичне програмне забезпечення з інтерфейсом ноутбука, програмованим на Python: його бібліотека охоплює багато аспектів математики , включаючи алгебру , комбінаторику , чисельну математику , теорію чисел і обчислення .

Пітон був успішно впроваджений у багатьох програмних продуктів, як скриптової мови, включаючи метод кінцевих елементів програмного забезпечення, такого як комплекс Aباqus, для 3D параметричного моделювання, як в FreeCAD, 3D анімації пакети, такі як 3DS Макс, блендер, кіно 4D, Lightwave обходить навколо, Гудіні, Майя, Модо, Мобі, програма, візуальні ефекти складач нюк, 2D-візуалізації програми, як GIMP, В Inkscape, Scribus виступає і фарба магазин професійний, і нотного запису програми, як нотний редактор і капели. Налагоджувача GNU Python використовує в якості симпатичного принтера для відображення складних структур, таких як контейнери C++. Esri просуває Python як кращий вибір для написання скриптів в ArcGIS . він також був використаний в декількох відеоіграх, і був прийнятий в якості

першого з трьох доступних мов програмування в Google App Engine , два інших-Java і Go .

Python зазвичай використовується в проектах з допомогою таких бібліотек, як TensorFlow , Keras, Pytorch і Scikit-learn . як скриптова мова з модульною архітектурою , простим синтаксисом та інструментами обробки насиченого тексту, Python часто використовується для обробки природної мови .

Багато операційні системи включають Python в якості стандартного компонента. Він поставляється з більшістю дистрибутивів Linux, AmigaOS 4 (з використанням Python 2.7), FreeBSD (у вигляді пакета), NetBSD , OpenBSD (у вигляді пакета) і macOS і може бути використаний з командного рядка (термінал). Багато дистрибутиви Linux використовують установники, написані на Python: Ubuntu використовує інсталятор Ubiquity, в той час як Red Hat Linux і Fedora використовують установник Anaconda. Gentoo Linux використовує Python у своїй системі управління пакетами Portage .

Python широко використовується в індустрії інформаційної безпеки, в тому числі при розробці експлойтів.

Велика частина програмного забезпечення для одного ноутбука на дитину XO, в даний час розробленого в Sugar Labs , написана на Python. одноплатний комп'ютерний проект прийняв Python у якості свого основного інтерфейсу мови програмування.

Завдяки зручному для користувача угодою Python і простим для розуміння мови, він зазвичай використовується в якості вступного мови в обчислювальних науках зі студентами.

LibreOffice включає Python і має намір замінити Java на Python. Його постачальник Python є основною функцією 3 версії 4.0 від 7 лютого 2013 року.

## Вибір мови для обробки зовнішнього вигляду

Для розробки зовнішньої оболонки було обрано мову CSS так як вона є найефективнішим вибором

**CSS** – це мова таблиці стилів, яка використовується для опису подання документа, що написаний мовою розмітки, наприклад, HTML.[8<sup>1</sup>] CSS є на-  
ружним каменем всесвітньої павутини технології, поряд з HTML і JavaScript.

Логотип мови CSS зображено на рисунку 5



Рисунок 5 – Логотип мови CSS

CSS призначено для розділення презентацій та вмісту, включно з макетом, кольорами та шрифтами [9]<sup>2</sup>. такий поділ може поліпшити доступність вмісту, забезпечити більшу гнучкість і контроль у специфікаціях продуктивності, і дозволити декільком веб-сторінок для обміну форматуванням, вказавши відповідний CSS в окремому. CSS файлі, і зменшити складність і повто-

---

<sup>1</sup> [8] "Руководство для разработчиков CSS" URL: <https://developer.mozilla.org/en-US/docs/Learn/CSS>

<sup>2</sup> [9] "Що таке CSS?" URL: <https://www.w3.org/standards/webdesign/htmlcss#whatcss>



рення в структурному вмісті. Поділ форматування та змісту також дає можливість представити однакову розмітку сторінки в різних стилях для різних візуалізацій методів, таких як екран, друк, голос (через мовлення на основі браузера або пристрою читання з екрану) і на основі Брайля пристроїв тактильної. CSS також має правила.

## Синтаксис мови CSS

CSS має простий синтаксис і використовує ряд англійських ключових слів для позначення назв різних властивостей стилю. Таблиця стилів складається зі списку правил. Кожне правило або набір правил складається з одного або кількох селекторів і блоків оголошень.

Блок оголошень складається зі списку оголошень в фігурних дужках. Кожне оголошення саме по собі складається з властивості, двокрапки (:) і значення. Якщо в блоці є кілька оголошень ;, то для поділу кожного оголошення необхідно вставити крапку з комою ([10]<sup>1</sup>).

Властивості задаються в стандарті CSS. Кожна властивість має набір можливих значень. Деякі властивості можуть впливати на будь-який тип елемента, а інші будуть застосовуватись тільки до певних груп елементів.

Значення можуть бути ключовими словами, такими як "центр" або "успадковувати", або числовими значеннями, такими як 200px (200 пікселів), 50vw (50 відсотків ширини видового екрану) або 80% (80 відсотків ширини батьківського елемента). Кольорові значення можуть бути задані за допомогою ключових слів (наприклад, "red"), шістнадцятирічних значень (наприклад # FF0000, також скорочено # F00), значень RGB в масштабі від 0 до 255 (наприклад rgb (255, 0, 0)), значень RGBA, що визначають колір і Альфа-прозорість (наприклад rgba (255, 0, 0, 0.8)), або значень HSL або HSLA (наприклад hsl (000, 100%, 50%), hsla (000, 100%, 50%, 80%)).

---

<sup>1</sup> [10] "Специфікація W3C CSS2. 1 для наборів правил, блоків оголошень і селекторів" URL: <https://www.w3.org/TR/CSS21/syndata.html#q10>

## Одиниці виміру довжини в CSS

Ненульові числові значення, що представляють лінійні заходи, повинні включати одиницю довжини, яка є або літерним кодом, або аббревіатурою, як в 200px і лі 50vw; або знаком відсотка, як всередині 80%. У деяких номерах – cm (см); in (дюймів); mm (мм); pc (пайка); а pt (точка) – це абсолют, який означає, що надані вимір не залежить від структури сторінки; інші – em (EM); ex (ex) і px (пікселів) – кілька відносно, це означає, що такі фактори, як розмір шрифту батьківського елемента може вплинути на винесене вимірювання.

Ці вісім одиниць були особливістю CSS 1 і зберігається в усіх наступних редакціях. Пропоновані значення CSS і одиниці модуля рівня 3, якщо вони будуть прийняті в якості рекомендації W3C, забезпечать сім додаткових одиниць довжини: ch; Q; rem; vh; vmaxvmin; і vw.

## Використання мови CSS

До появи CSS майже всі презентаційні атрибути HTML-документів містилися в розмітці HTML. Всі кольори шрифтів, стилі фону, вирівнювання елементів, межі і розміри повинні були бути явно описані, часто повторно, в HTML. CSS дозволяє авторам перенести більшу частину цієї інформації в інший файл, таблицю стилів, що призводить до значно більш простому HTML.

Наприклад, заголовки (h1 елементи), підзаголовки (h2), підзаголовки (h3) і т. Д., Визначаються структурно з використанням HTML. У пресі і на екрані вибір шрифту, розміру, кольору і акценту для цих елементів є презентаційним.

До CSS, автори документа, які хотіли привласнити такі друкарські характеристики, скажімо, всім h2 заголовки повинні були повторювати HTML презентаційну розмітку для кожного входження цього типу заголовка. Це

зробило документи більш складними, великими, схильними до помилок і важкими в обслуговуванні. CSS дозволяє відокремити презентацію від структури. CSS може визначати колір, шрифт, вирівнювання тексту, розмір, межі, інтервал, макет і багато інших друкарських характеристик, а також може робити це незалежно для екранних і друкованих уявлень. CSS також визначає невізуальні стилі, такі як швидкість читання і акцент для звукових текстових читачів. W3C тепер не рекомендується використовувати всю презентаційну розмітку HTML.[11]<sup>1</sup>

Наприклад, в розділі pre-CSS HTML елемент заголовка, певний червоним текстом, буде записаний як:

```
<h1> <color = "red"> Chapter 1. </ font> </ h1>
```

Використовуючи CSS, один і той же елемент можна закодувати за допомогою властивостей стилю замість атрибутів уявлення HTML:

```
<h1 style = "color: red;"> Chapter 1. </ h1>
```

Переваги цього можуть бути не відразу зрозумілі, але сила CSS стає більш очевидною, коли властивості стилю розміщуються у внутрішньому елементі стилю або, ще краще, в зовнішньому файлі CSS. Наприклад, припустимо, що документ містить елемент style:

```
<Style> h1 {color: red;} </ Style>
```

Все h1 елементи в документі потім автоматично стануть червоними, не вимагаючи ніякого явного коду. Якщо автор пізніше хотів зробити h1 елементи синіми замість цього, це можна було б зробити, змінивши елемент стилю на:

```
<Style> h1 {color: blue;} </ Style>
```

замість того, щоб ретельно проходити через документ і змінювати колір для кожного окремого h1 елемента.

Стилі також можуть бути поміщені у зовнішній файл CSS, як описано нижче, і завантажені з використанням синтаксису, аналогічного:

---

<sup>1</sup> [11] "HTML 5. Словник і пов'язані з ним API для HTML і XHTML" URL: <https://html.spec.whatwg.org/multipage/introduction.html#presentational-markup>

```
<Link href = "path / to / file.css" rel = "stylesheet" type =  
"text / css">
```

Це додатково відокремлює стиль від HTML-документа і дозволяє змінити стиль декількох документів, просто відредагувавши загальний зовнішній файл CSS.

## Спадкування в мови CSS

Спадкування є ключовою функцією в CSS; він покладається на предок-нащадок відносини для роботи. Спадкування-це механізм, за допомогою якого властивості застосовуються не тільки до зазначеного елемента, але і до його нащадкам. спадкування ґрунтується на дереві документів, яке є ієрархією XHTML елементи на сторінці, засновані на вкладеності. Елементи-нащадки можуть успадковувати значення властивостей CSS від будь-якого елемента-предка, що містить їх. Як правило, елементи-нащадки успадковують властивості, пов'язані з текстом, але їх властивості, пов'язані з полем, не успадковуються. Властивості, які можуть бути успадковані: колір, шрифт, інтервал між літерами, висота рядка, стиль списку, вирівнювання тексту, відступ тексту, перетворення тексту, видимість, пробіл і інтервал між словами. Властивості, які не можуть бути успадковані, – це фон, межа, відображення, плаваючий і ясний, висота і ширина, поле, мінімальна і максимальна висота і ширина, контур, переповнення, заповнення, положення, оформлення тексту, вертикальне вирівнювання та Z – індекс.

Спадкування може використовуватися, щоб уникнути оголошення певних властивостей знову і знову в таблиці стилів, що дозволяє скоротити CSS.

Спадкування в CSS – це не те ж саме, що спадкування в класових мовах програмування, де можна визначити клас B як "подібний класу A, але з модифікаціями". За допомогою CSS можна стилізувати елемент з "класом A, але з модифікаціями". Проте, неможливо визначити клас CSS B подібним чи-

ном, який потім можна було б використовувати для стилізації декількох елементів без необхідності повторювати зміни.

## **Варіації версії мови CSS**

CSS має різні рівні та профілі. Кожен рівень CSS будується на останньому, зазвичай додаючи нові функції і зазвичай позначається як CSS 1, CSS 2, CSS 3 та CSS 4. Профілі, як правило, представляють собою підмножина одного або декількох рівнів CSS, побудованих для конкретного пристрою або призначеного для користувача інтерфейсу. В даний час існують профілі для мобільних пристроїв, принтерів і телевізорів. Профілі не слід плутати з типами носіїв, які були додані в CSS 2.

## **Характеристика мови CSS версії 1**

Першою специфікацією CSS, яка стала офіційною рекомендацією W3C, є CSS level 1, опублікована 17 грудня 1996 року. Нанон Wium Lie і Bert Bos вважаються оригінальними розробниками. серед його можливостей є підтримка для:

- Властивості шрифту, такі як шрифт і акцент;
- Колір тексту, фону і інших елементів;
- Атрибути тексту, такі як інтервал між словами, літерами та рядками тексту;
- Вирівнювання тексту, зображень, таблиць і інших елементів;
- Поля, кордони, відступи і позиціонування для більшості елементів;
- Унікальна ідентифікація та загальна класифікація груп ознак.

W3C більше не підтримує рекомендацію CSS 1.

### **2.2.5.2 Характеристика мови CSS версії 2**

Специфікація CSS level 2 була розроблена W3C і опублікована в якості рекомендації в травні 1998 року. Надбезліччю для CSS 1 і CSS 2 включає в себе ряд нових можливостей, як абсолютні, відносні і фіксоване позиціонування елементів і Z-індекс, поняття типів носіїв, підтримка звукових таблиць стилів (які згодом були замінені на CSS 3 мовних модулів) і двонаправленого тексту, і шрифт властивості, такі як тіні.

W3C більше не підтримує рекомендацію CSS 2.

### **2.2.5.3 Характеристика мови CSS версії 2.1**

CSS level 2 revision 1, Часто званий "CSS 2.1", виправляє помилки в CSS 2, видаляє погано підтримувані або не повністю сумісні функції і додає в специфікацію вже реалізовані браузерні розширення. Щоб відповідати процесу W3C по стандартизації технічних специфікацій, CSS 2.1 протягом багатьох років переходив від робочого проекту статусу до статусу рекомендації кандидата. CSS 2.1 вперше став кандидатом рекомендація 25 лютого 2004 року, але він був повернутий до робочого проекту 13 червня 2005 для подальшого розгляду. Він повернувся до рекомендації щодо кандидатів 19 липня 2007 року потім двічі оновлювався в 2009 році. Однак, оскільки були внесені зміни і уточнення, він знову повернувся до робочого проекту Last Call 7 грудня 2010 року.

CSS 2.1 перейшов до запропонованої рекомендації 12 квітня 2011 року. після розгляду Консультативним комітетом W3C він був остаточно опублікований в якості рекомендації W3C 7 червня 2011 року.

CSS 2.1 був запланований в якості першого і остаточного перегляду рівня 2-но в 2015 році почалася фонові робота над CSS 2.2.

### **2.2.5.2 Характеристика мови CSS версії 2**

"CSS3" перенаправляє сюди. Для інших цілей см. CSS3 (disambiguation).

На відміну від CSS 2, який представляє собою велику єдину специфікацію, визначальну різні функції, CSS 3 розділений на кілька окремих документів, які називаються "модулями". Кожен модуль додає нові можливості або розширює можливості, певні в CSS 2, зберігаючи сумісність. Робота над CSS level 3 почалася приблизно в той же час, коли була опублікована оригінальна рекомендація CSS 2. Найбільш ранні проекти CSS 3 були опубліковані в червні 1999 року

Належний до модуларизації, різні модулі мають різні стабільність і стану.

Деякі модулі мають статус рекомендацій-кандидатів (CR) і вважаються помірно стабільними. На етапі CR реалізації рекомендується відкинути префікси постачальників.

### **2.2.5.5 Характеристика мови CSS версії 4**

Там немає єдиної, інтегрованої специфікації CSS4, тому що специфікація була розділена на багато окремих модулів, які вирівнюють незалежно.

Модулі, які будують на речах з CSS рівня 2, почалися на рівні 3. Деякі з них вже досягли рівня 4 або вже наближаються до рівня 5. Інші модулі, які визначають абсолютно нову функціональність, такі як Flexbox, були позначені як рівень 1, і деякі з них наближаються до рівня 2.

Робоча група CSS іноді публікує "моментальні знімки", колекцію цілих модулів і частин інших проектів, які вважаються досить стабільними для реалізації розробниками браузерів. До теперішнього часу було опубліковано п'ять таких документів "найкраща поточна практика" у вигляді приміток, зокрема: 2007, 2010, 2015 2017, і в 2018 році.

Оскільки ці знімки специфікації призначені в першу чергу для розробників, спостерігається зростаючий попит на аналогічний версійність довідковий документ, призначений для авторів, який представляв би стан взаємодіючих реалізацій, як це було задокументовано сайтами, такими як Can I Use... і Mozilla Developer Network. на початку 2020 року було створено групу спільноти W3C для обговорення та визначення такого ресурсу. фактичний вид контролю версій також знаходиться на стадії обговорення, що означає, що документ, одного разу підготовлений, не може називатися "CSS4".

### **Підтримка веб-браузерів**

Додаткова інформація: порівняння движків веб-браузера (підтримка CSS)

Кожен веб-браузер використовує механізм компонування для візуалізації веб-сторінок, і підтримка функціональності CSS не є узгодженою між ними. Оскільки браузери не аналізують CSS ідеально, було розроблено кілька методів кодування для цільових конкретних браузерів з обхідними шляхами (зазвичай відомі як CSS-хакі або CSS-фільтри). Ухвалення нової функціональності в CSS може бути ускладнене відсутністю підтримки в основних браузерах. Наприклад, Internet Explorer повільно додавав підтримку для багатьох функцій CSS 3, що уповільнювало прийняття цих функцій і ушкоджувало репутацію браузера серед розробників. Щоб забезпечити узгоджений досвід для своїх користувачів, веб-розробники часто тестують свої сайти в декількох операційних системах, браузерах і версіях браузерів, збільшуючи час розробки і складність. Такі інструменти, як BrowserStack, були створені для зниження складності обслуговування цих середовищ.

На додаток до цих інструментів тестування, багато сайтів підтримують списки підтримки браузерів для конкретних властивостей CSS, в тому числі CanIUse і Mozilla Developer Network. Крім того, CSS 3 визначає запити функцій, які надають @ supportsдирективу, яка дозволить розробникам



націлювати браузер з підтримкою певних функцій безпосередньо в їх CSS. CSS, яка не підтримується більш старими браузерами, також іноді може бути виправлений за допомогою поліфіллів JavaScript, які представляють собою фрагменти коду JavaScript, призначені для забезпечення узгодженої поведінки браузерів. Ці обхідні шляхи – а також необхідність підтримки резервної функціональності – можуть ускладнити проекти розробки, і, отже, компанії часто визначають список версій браузера, які вони будуть і не будуть підтримувати.

Оскільки веб-сайти приймають нові стандарти коду, несумісні зі старими браузерами, ці браузери можуть бути відрізані від доступу до багатьох ресурсів в інтернеті (іноді навмисно). багато з найпопулярніших сайтів в Інтернеті не просто візуально деградують на старих браузерах через погану підтримку CSS, але і не працюють взагалі, в значній мірі через еволюції JavaScript та інших веб-технологій.

### **Можливості мови CSS**

Коли CSS використовується ефективно, з точки зору успадкування і "каскадирования", глобальна таблиця стилів може використовуватися для впливу і стильові елементи на рівні сайту. Якщо виникає ситуація, що стиль елементів повинен бути змінений або скоректований, ці зміни можуть бути внесені шляхом редагування правил в глобальній таблиці стилів. До CSS, цей вид обслуговування був більш важким, дорогим і вимагає багато часу.

Таблиця стилів, внутрішній або зовнішній, визначає стиль один раз для діапазону HTML-елементів, обраних по класу або зв'язку з іншими. Це набагато ефективніше, ніж повторювати інформацію про стилі, вбудовану для кожного входження елемента. Зовнішня таблиця стилів зазвичай зберігається в кеші браузера і тому може використовуватися на декількох сторінках без перезавантаження, що ще більше скорочує передачу даних по мережі.

При простому зміні одного рядка для однієї і тієї ж сторінки можна використовувати іншу таблицю стилів. Це має переваги для доступності, а також надає можливість адаптувати сторінку або сайт до різних цільовим пристроїв. Крім того, пристрої, які не здатні зрозуміти стиль, як і раніше відображають вміст.

Без CSS веб-дизайнери зазвичай мають розміщувати свої сторінки за допомогою таких методів, як таблиці HTML, які ускладнюють доступність для користувачів з вадами зору (див. [Tableless web design # Accessibility](#)).

### **Середовище розробки зовнішнього вигляду QT**

Для створення графічної оболонки було обрано програму QT

Qt є вільним і відкритим вихідним кодом віджета інструментарій для створення графічних користувацьких інтерфейсів, а також крос-платформних додатків, які працюють на різних програмних і апаратних платформах, таких як ОС Linux, Windows, ОС macOS, Андроїд або вбудованих систем. З невелике або ніяка зміна у вихідному коді і в той же час рідне додаток з власним можливостям і швидкості

Логотип програми для роботи з зовнішню оболонкою QT зображено на рисунку 6.



Рисунок 6 – Логотип програми Qt

Qt в даний час розробляється компанією Qt, публічно котируваною компанією, і проектом Qt під управлінням з відкритим вихідним кодом, з залученням індивідуальних розробників і організацій, що працюють над просуванням Qt[12]<sup>1</sup>. Qt доступний як під комерційними ліцензіями, так і під ліцензіями з відкритим вихідним кодом GPL 2.0, GPL 3.0 і LGPL 3.0. Зовнішній вигляд програми QT зображений на рис.7.

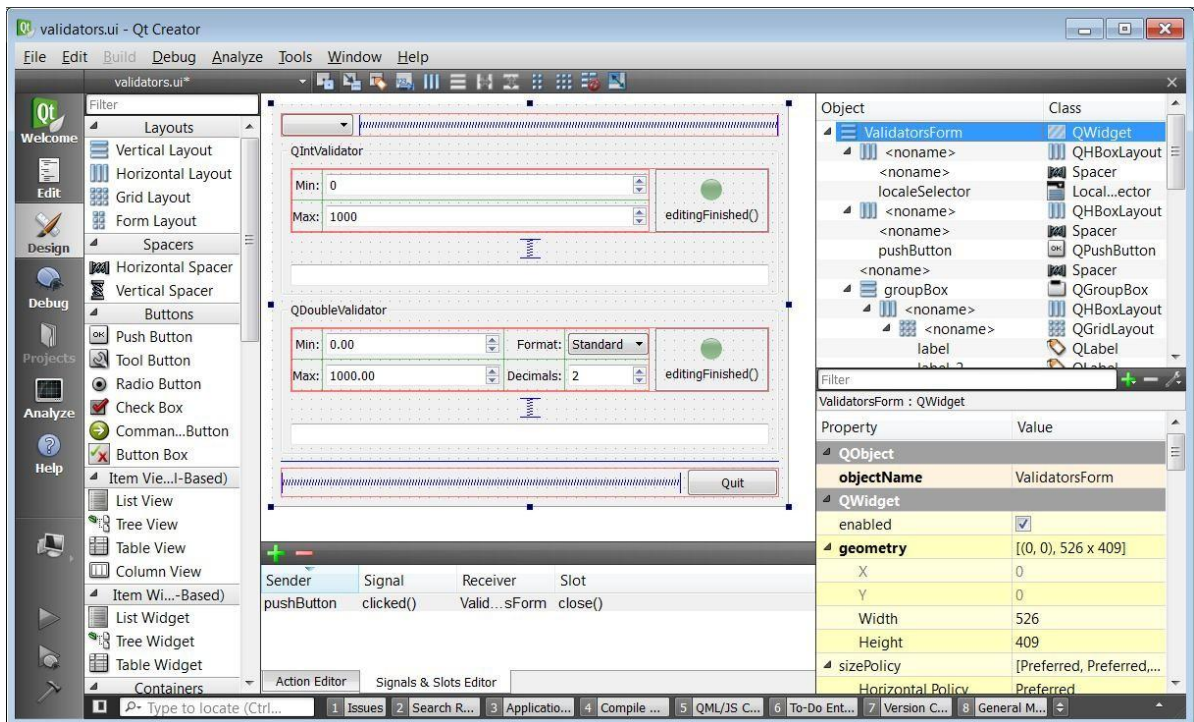


Рисунок 7 – зовнішній вигляд програми QT

Qt використовується для розробки графічних інтерфейсів користувача (GUI) і мультиплатформових додатків, які працюють на всіх основних настільних платформах і більшості мобільних або вбудованих платформ. Більшість графічних програм, створених за допомогою Qt, мають нативний інтерфейс, і в цьому випадку Qt класифікується як інструментарій віджетів. Також можуть бути розроблені програми без графічного інтерфейсу користу-

<sup>1</sup> [12] Meyer, David (24 Жовтня 2011). "Nokia дає Qt управління з відкритим вихідним кодом" URL: <https://www.zdnet.com/article/nokia-gives-qt-open-source-governance/>

вача, такі як засоби командного рядка і консолі для серверів. Прикладом такої не-графічної програми, що використовує Qt, є Cutelyst web framework.

Qt підтримує різні компілятори, в тому числі компілятор GCC C++ і Visual Studio suite, і має велику підтримку інтернаціоналізації. Qt також надає Qt Quick, який включає в себе декларативний мову сценаріїв, званий QML, який дозволяє використовувати JavaScript для забезпечення логіки. З Qt Quick стала можливою швидка розробка додатків для мобільних пристроїв, в той час як логіка все ще може бути написана з використанням власного коду, а також для досягнення найкращої продуктивності.

Інші функції включають доступ до бази даних SQL, синтаксичний аналіз XML, аналіз JSON, управління потоками і мережеву підтримку.

Архітектура програмного забезпечення Qt зображена на рис.8

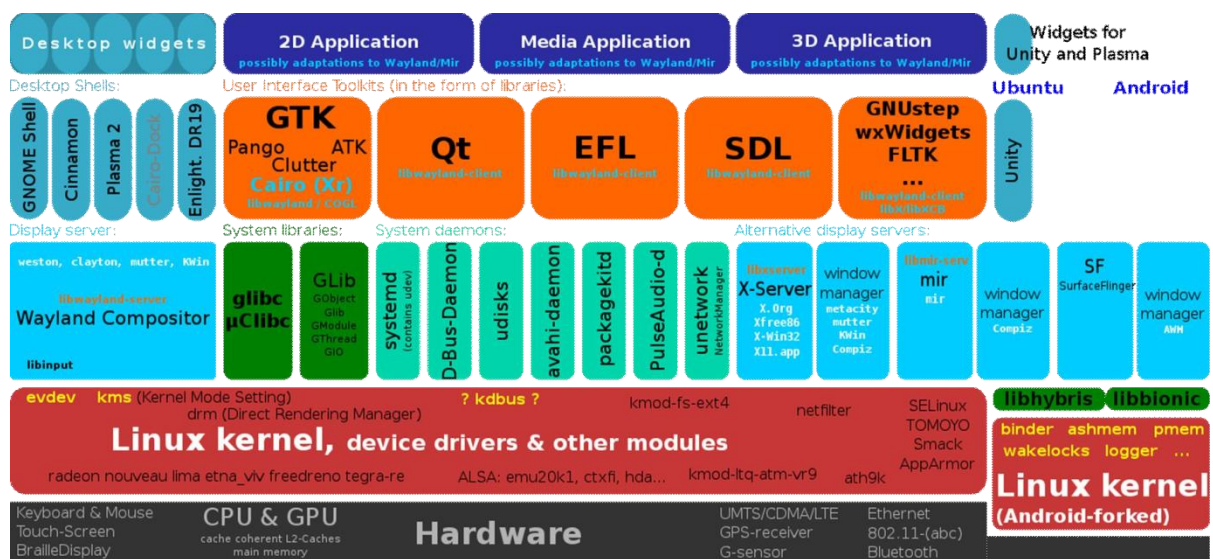


Рисунок 8 – Приклад використання Qt у системах на базі Linux

## Концепції програми Qt

Повна абстракція графічного інтерфейсу користувача. При першому випуску Qt використала свій власний движок малювання і елементи управління, емулюючи зовнішній вигляд різних платформ, на яких вона

працює, коли вона малювала свої віджети. Це полегшувало роботу з портированием, оскільки дуже небагато класи в Qt залежали насправді від цільової платформи; однак це іноді призводило до невеликим розходженням там, де ця емуляція була недосконалою. Останні версії Qt використовують API власного стилю різних платформ, на платформах, які мають власний набір віджетів, щоб запитувати метрики і малювати більшість елементів керування, і не страждають від таких проблем, як часто. На деяких платформах (таких як MeeGo і KDE) Qt є власний API. Деякі інші портативні графічні набори інструментів прийняли різні проектні рішення; наприклад, wxWidgets використовує набори інструментів цільової платформи для своїх реалізацій.

Мовна конструкція, введена в Qt для зв'язку між об'єктами, яка дозволяє легко реалізувати шаблон спостерігача, уникаючи шаблонного коду. Концепція полягає в тому, що віджети GUI можуть надсилати сигнали, які містять інформацію про події, які можуть бути отримані іншими елементами керування за допомогою спеціальних функцій, відомих як слоти.

Компілятор `metaobject`, званий `moc`, є інструментом, який запускається на джерелах програми Qt. Він інтерпретує певні макроси з коду C++ як анотації та використовує їх для створення додаткового коду C++ з метаінформацією про класах, що використовуються в програмі. Ця метаінформація для використовується Qt для надання функцій програмування, не доступних спочатку в C++: сигнали і слоти, інтроспекція і асинхронні виклики функцій.

Qt може використовуватися в декількох інших мовах програмування, таких як Python, Javascript, C# або Rustз допомогою мовних Прив'язок, див.

Починаючи з Qt 4.0 фреймворк був розділений на окремі модулі. З Qt архітектура була модулірована ще більше. Qt тепер розділений на основні і додаткові модулі[13]<sup>1</sup> які зображені у таблиці 2.

---

<sup>1</sup> [13] "QtDoc 5.1: Всі Модулі" URL: <https://doc.qt.io/qt-5/qtmodules.html>

Таблиця 2 – Qt модулі

Модуль	Опис
1	2
Qt Core	Єдиний необхідний модуль Qt, що містить класи, що використовуються іншими модулями, включаючи метаоб'єктну систему, паралелізм та потокову обробку, контейнери, систему подій, плагіни і засоби вводу-виводу.
Qt Widgets	Містить класи для класичних додатків на основі віджетів GUI і класів QSceneGraph. Був відділений від QtGui в Qt 5.
Qt QML	Модуль для мов QML і JavaScript.
Qt Quick	Модуль для графічного інтерфейсу програми написаний з використанням QML2.
Qt Quick Controls	Віджет як елементи керування для Qt Quick призначений в основному для настільних додатків.
Qt Quick Layouts	Макети для організації елементів у Qt Quick.
Qt Network	Рівень абстракції мережі. У комплекті з підтримкою TCP, UDP, HTTP, SSL і починаючи з Qt 5.3 SPDY.
Qt Multimedia	Класи для аудіо, відео, радіо і функціональності камери.
Qt Multimedia Widgets	Віджети з Qt Multimedia.
Qt SQL	Містить класи для інтеграції баз даних за допомогою SQL.
Qt WebEngine	Новий набір Qt Widget і QML webview API, заснованих на Chromium
Qt Test	Класи для модульного тестування додатків і бібліотек Qt.

### Видання програми Qt

Існує чотири випуски Qt: Community, Indie Mobile, Professional і Enterprise версія спільноти знаходиться під ліцензіями з відкритим вихідним кодом, в той час як версії Indie Mobile, Professional і Enterprise, які містять додаткову функціональність і бібліотеки, наприклад Enterprise Controls, комерційно продаються компанією Qt.

## Підтримувані платформи Qt

Qt працює на різних платформах; офіційно підтримуються наступні компоненти дивитися у таблиці 3

Таблиця 3 – Платформах яки придтримують Qt

Платформа	Опис
1	2
Linux / Unix	
X11	Qt для X Window System (Linux); FreeBSD , NetBSD , OpenBSD і DragonFly BSD мають підтримку спільноти.
Wayland	Qt для Wayland (наприклад, SailfishOS використовує це, оскільки у нього немає X11) додатки Qt можуть перемикатися між графічними бэкендами, такими як X і Wayland, під час завантаження з опцією командного рядка-platform. це дозволяє плавно переходити Qt-застосунків X11 в Wayland.
Android	Qt для Android (раніше відомий як Necessitas).
Вбудований Linux	Qt для вбудованих платформ: персональний цифровий помічник, смартфон і ін існує декілька платформ в залежності від технології відображення. DirectFB, LinuxFB і EGLFS (повний екран EGL).
платформа Microsoft	
Windows	Qt для Microsoft Windows 7, 8 і 10
Windows RT	Підтримка мобільних додатків на базі WinRT для Windows 10 і Windows 10 IoT

Платформи Apple	
МАКО	Qt для Apple macOS; підтримує програми на Cocoa
iOS	Qt для платформ iOS (iPhone, iPad )
Інші вбудовані платформи	
Цілісність	Qt для цілісності
компанія QNX	Qt для QNX
VxWorks	Qt для VxWorks тільки доступний під собственической (комерційно) ліцензією. Qt 5.5.

Після того, як Nokia відкрила вихідний код Qt для спільноти на Gitorious з'явилися різні порти. Є також деякі порти Qt, які можуть бути доступні, але більше не підтримуються. Ці платформи перераховані в списку

платформ, підтримуваних Qt . Дивіться також там для поточної підтримки спільноти для інших менш відомих платформ, таких як SailfishOS.

### Ліцензування програмного забезпечення

Qt доступний під такими ліцензіями на вільне програмне забезпечення: GPL 2.0 , GPL 3.0 , LGPL 3.0 і LGPL 2.1 [14]<sup>1</sup> (за винятком Qt special). Деякі модулі доступні тільки за ліцензією GPL, що означає, що програми, які статично пов'язані з цими модулями, повинні відповідати цій ліцензії.

Крім того, Qt завжди була доступна під комерційною ліцензією, такий як комерційна ліцензія Qt, яка дозволяє розробляти власні додатки без обмежень на ліцензування.

### Інструменти Qt

Qt поставляється зі своїм власним набором інструментів для полегшення крос-платформної розробки, яка в іншому випадку може бути громіздкою з-за різного набору інструментів розробки дивитись на рис.9.

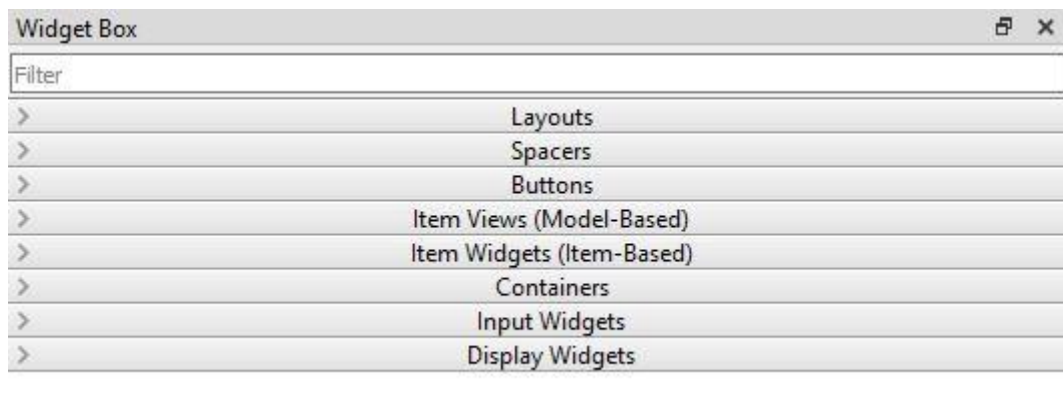


Рисунок 8 – набором інструментів

<sup>1</sup>[14] Qt. "Legal-FAQ-Qt" URL: <https://www.qt.io/faq/>



Qt Creator-це платформна IDE C++ і QML. Функціональність компонування/дизайну Qt GUI Designer інтегрована в IDE, хоча Qt Designer все ще може бути запущений як автономний інструмент.

На додаток до Qt Creator, Qt надає qmake , крос-платформний інструмент для генерації скриптів збірки, який автоматизує генерацію файлів Makefile для розробки проектів на різних платформах. Є й інші інструменти, доступні в Qt, включаючи Qt Designer interface builder і Qt Assistant help browser (які обидва вбудовані в Qt Creator), інструмент перекладу Qt Linguist, uic (User interface compiler) і moc (Meta-Object Compiler).

## РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ЧАСТИНИ

### Створення аудіо програвоча

Для роботи з аудіо файлами було обрано бібліотеку пайтон Pygame та Mutagen які дозволяють працювати з аудіо файлами Pygame для роботи з аудіо модулі призначені `pygame.mixer` і `pygame.mixer.music`. Модулі схожі, однак `pygame.mixer` в першу чергу адаптований для додавання та налаштування звукових ефектів в грі. У той час як `pygame.mixer.music` – для додавання фонові музики.

Функція `pygame.mixer.music.load()` завантажує потокове аудіо, тобто не вантажить файл цілком, а робить це окремими порціями. В результаті можна програвати тільки один файл за раз. Однак можна ставити файли в чергу функцією `queue()`. Підтримує в тому числі формат `mp3`.

З іншого боку, в `pygame.mixer` ключовим є клас `Sound`. Він дозволяє завантажувати, відтворювати і виконувати ряд інших дій з файлами форматів `wav` або `ogg`. При створенні екземпляра `Sound` в конструктор передається ім'я файлу.

За допомогою функції `music.play()` файл починає програватися. Якщо потрібно зациклити композицію, то в `play()` передається число `-1`. Позитивний аргумент вказує на кількість повторів `+` одне додаткове. Тобто, якщо треба програти композицію 2 рази, то в функцію передається число `1`.

У програмі при натисканні на клавішу `PAUSE` клавіатури музика ставиться на паузу: `music.pause()`. Клавіша `MOOT` зменшує гучність в два рази: `music.set_volum(0)`. Натискання `MOOT` повертає гучність на колишній рівень. Функція `unpause()` викликається на випадок, якщо до цього музика була вимкнена (`STOP`).

`Mutagen` – це модуль Python для обробки аудіо метаданих. Він підтримує `ASF`, `FLAC`, `MP4`, аудіо мавпи, `MP3`, `Musepack`, `Ogg Opus`, `Ogg FLAC`, `Ogg Speex`, `Ogg Theora`, `Ogg Vorbis`, `True Audio`, `WavPack`, `OptimFROG`

і AIFF аудіофайли. Всі підтримуються версії ID3v2, і всі стандартні ID3v2 кадри.4 аналізуються. Він може читати заголовки Xing, щоб точно розрахувати швидкість і довжину Теги MP3s. ID3 і APEv2 можна редагувати незалежно від формату аудіо. Він також може управління потоками Ogg на рівні окремого пакета/сторінки.

Мутаген працює з Python 3.5+ (CPython і PyPy) на Linux, Windows і macOS, і не має ніяких залежностей за межами стандартної бібліотеки Python. Мутаген ліцензується під GPL версії 2 або більш пізньої.

Quod Libet має більш напружені вимоги в бібліотеці тегів, ніж більшість програми, які мають справу з тегами. Тому ми вважали за необхідне написати наші власний.

- Мутаген має простий API, який приблизно однаковий у всіх форматах тегів а також версії і інтегрується в вбудовані типи і інтерфейси Python Ст;

- Нові типи фреймів і формати файлів легко додаються, а поведінку поточні формати можуть бути змінені шляхом їх розширення.

- Ключі Freeform, множинні значення, Unicode, і інші попередні характеристики були розглядаються з самого початку і повністю підтримуються;

- Всі версії ID3v2 і всі рамки ID3v2.4 покриті, включаючи рідкісні одні як ПОПМ або RVA2;

Ми дуже серйозно ставимося до автоматизованого тестування. Всі виправлення помилок фіксуються з допомогою тест, який запобігає їх повторення, і нові функції фіксуються з допомогою повний набір тестів.

Мутаген може завантажувати майже кожен MP3, який ми кинули на нього (коли це не так, ми примусьте його зробити це). Сценарії включені, так що ви можете запусити ті ж тести на вашому Колекція.

Такі проекти програмного забезпечення використовують Мутаген для маркування аудіо.

## Створення списку аудио файлів

Для створення списку аудио файлів було обрано бібліотеку OS та віджет PyQt-QFileDialog.

OS це бібліотека функцій для роботи з операційною системою. Методи, включені в неї дозволяють визначати тип операційної системи, отримувати доступ до змінних оточення, управляти директоріями і файлами:

- перевірка існування об'єкта по заданому шляху;
- визначення розміру в байтах;
- видалення;
- перейменування та ін.

OS можуть застосовуватися користувачем для різних цілей. Нижче наведені найбільш популярні з них, що дозволяють отримувати дані про операційній системі. Також отримувати відомості про файли та папки, що зберігаються в пам'яті на жорсткому диску ПК.

Щоб уникнути помилок, пов'язаних з відсутністю певного файлу або директорії, які повинні бути оброблені програмою, слід попередньо перевіряти наявність допомогою методу exists. Передавши йому в якості аргументу шлях до потрібного файлу або папки, можна розраховувати на лаконічну відповідь у вигляді булевого значення true/false, який повідомляє про наявність/відсутність зазначеного об'єкта в пам'яті комп'ютера. У наступному прикладі йде перевірка текстового файлу test.txt з кореневого каталогу D, яка повертає True.

```
import os
print(os.path.exists("D:/test.txt"))
```

виводе результат процесу – True

Якщо об'єкт на диску реально існує, це не завжди означає, що він має відповідну для подальшої обробки форму. Перевірити, чи є певний об'єкт файлом, допоможе функція isfile, яка приймає його адресу. Ознайомитися з

результатом його роботи можна з наступного прикладу, де `print` відображає на екрані значення `True` для файлу `test.txt`.

```
import os
print(os.path.isfile("D:/test.txt"))
```

виводе результат процесу – `True`

Аналогічні дії можна виконати і для перевірки об'єкта на приналежність до класу директорій, викликавши для його адреси метод `isdir` з бібліотеки `os`. Як можна помітити, у цьому випадку `print` виводить на екран логічне значення `False`, оскільки `test.txt` не є текою.

```
import os
print(os.path.isdir("D:/test.txt"))
```

виводе результат процесу – `False`

для взаємодії з документом необхідно отримати його повне ім'я, що включає дозвіл, але не абсолютний шлях до нього на диску. Перетворити адресу об'єкта в назву дозволяє функція `basename`, яка міститься в підмодулі `path` з бібліотеки `os`. Таким чином, наступний приклад показує перетворення шляху `test.txt` у просте ім'я файлу.

```
import os
print(os.path.basename("D:/test.txt"))
```

виводе результат процесу – `test.txt`

Зворотна ситуація виникає тоді, коли користувачеві потрібно отримати тільки шлях до файлу, без самої назви об'єкта. Це допоможе зробити метод `dirname`, який повертає шлях до заданого документом у строковому поданні, як це продемонстровано в невеликому прикладі нижче. Тут `print` виводить на екран адресу текстового документа в папці `folder`.

```
import os
print(os.path.dirname("D:/folder/test.txt"))
```

виводе результат процесу – `D:/folder`

Віджет `PyQt-QFileDialog` являє собою діалогове вікно вибору файлів. Він дозволяє користувачеві переміщатися по файловій системі і вибирати файл для відкриття або збереження. Діалогове вікно викликається або за до-

помогою статичних функцій, або за допомогою виклику функції `exec_()` для об'єкта діалогового вікна зображено на рис.10.

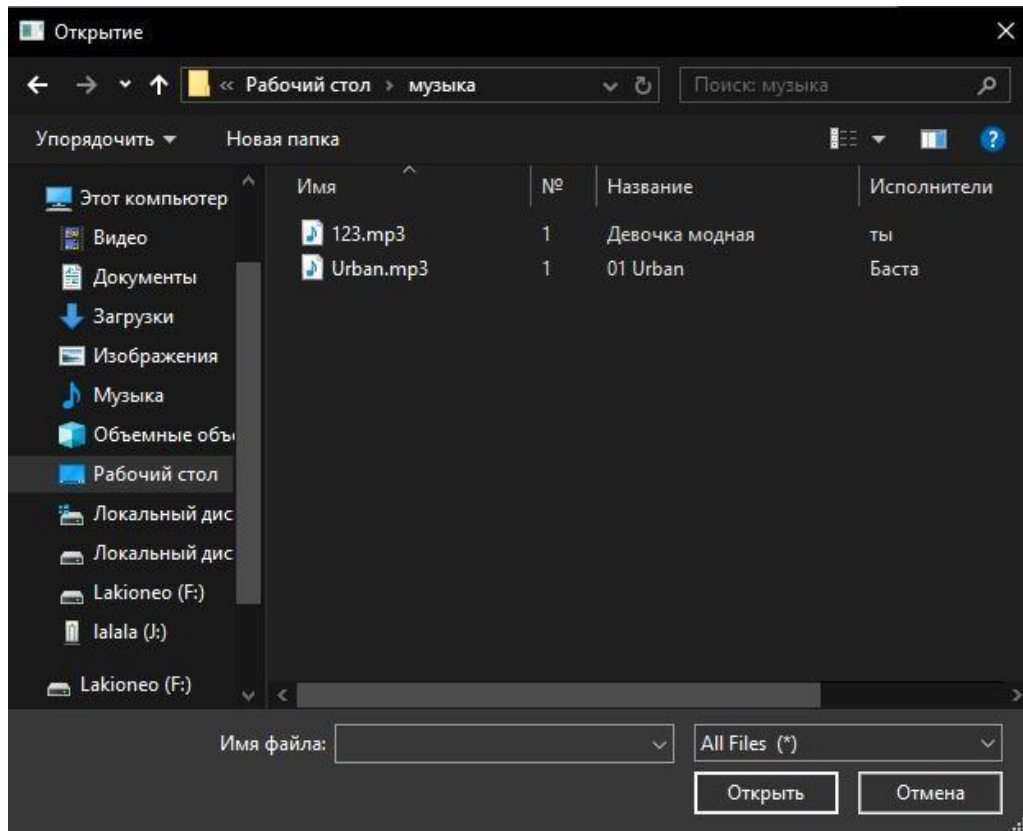


Рисунок 10 – Діалогове вікно для вибору аудіо файлу

Статичні функції класу `QFileDialog` (`getOpenFileName ()` і `getSaveFileName ()`) викликають власний файловий діалог поточної операційної системи.

Фільтр файлів також може застосовуватися для відображення файлів зазначених розширень. Також можна задати початковий каталог та ім'я файлу за замовчуванням.

який при обранні файлу записував його до 2 масивів. Один зберігав назву аудіо файлу та призначає йому індекс а другий записував повний шлях до аудіо файлу, це дає нам можливість робити перелік усіх аудіо файлів та програвати їх коли нам завгодно.

# выбор файла

```

def browse_file():
    global fill_path
    fill_path = QFileDialog.getOpenFileName()[0]
    add_to_playlist(fill_path)
# индекс
def add_to_playlist(filename):
    filename = os.path.basename(filename)
    index = 0
    ui.listWidget.insertItem(index, filename)
    playlist.insert(index, fill_path)
    index += 1

```

### Розробка зовнішнього вигляду програвача

Для створення зовнішнього вигляду була взята програма PyQT5 яка дозволяє створити програмне вікно на якому було розміщено один label один listWidget шість pushButton та два Slidr вони були розміщені на об'єкті Form.рис.11

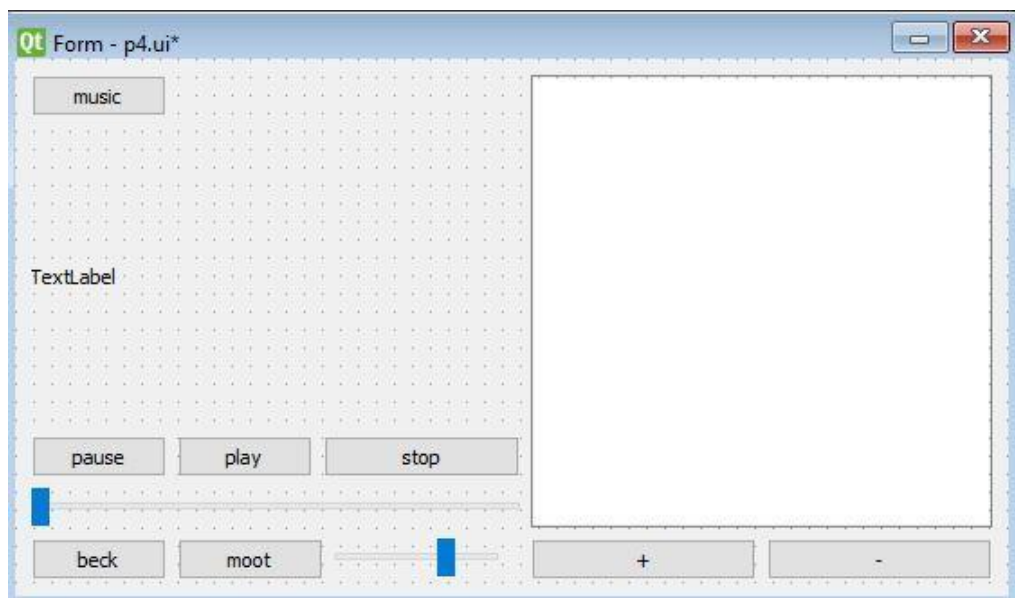


Рисунок 11 – розміщення об'єктів на панелі.

Після чого було створено з допомогою CSS зовнішній вигляд програми  
рис.12



Рисунок 12 – зовнішній вигляд програвач.

Для створення фону був створений на CSS клас QWidget

```
.QWidget {
color: #404040;
background: #373939;
}
```

При створенні зовнішнього вигляду кнопок були створені можливості взаємодії hover та pressed

Код CSS для кнопки рис.12

```
.QPushButton {
height: 48px;
line-height: 46px;
padding: 0 25px;
font-family: inherit;
font-size: 15px;
color: #bbb;
text-align: center;
text-decoration: none;
text-shadow: 0 0 2px rgba(0, 0, 0, 0.7);
}
```



```

background-color: #303030;
background-clip: padding-box;
border: 1px solid;
border-color: #202020 #1a1a1a #111;
border-radius: 25px;
background-image: -webkit-linear-gradient(top, #3d3d3d, #272727);
background-image: -moz-linear-gradient(top, #3d3d3d, #272727);
background-image: -o-linear-gradient(top, #3d3d3d, #272727);
background-image: linear-gradient(to bottom, #3d3d3d, #272727);
-webkit-box-shadow: inset 0 1px rgba(255, 255, 255, 0.09), 0 1px
3px rgba(0, 0, 0, 0.3);
box-shadow: inset 0 1px rgba(255, 255, 255, 0.09), 0 1px 3px
rgba(0, 0, 0, 0.3);
}

```

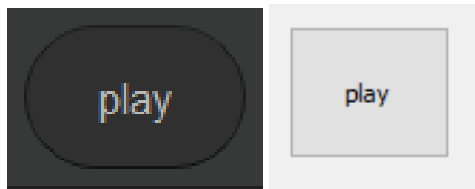


Рисунок 12– вигляд кнопки Play з CSS та бзе

Код Css для кнопки при наведенні рис.13

```

.QPushButton:hover {
background-color: #363636;
background-image: -webkit-linear-gradient(top, #404040, #2a2a2a);
background-image: -moz-linear-gradient(top, #404040, #2a2a2a);
background-image: -o-linear-gradient(top, #404040, #2a2a2a);
background-image: linear-gradient(to bottom, #404040, #2a2a2a);
}

```

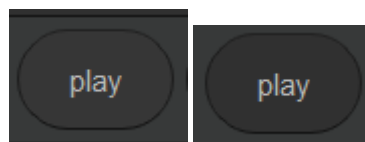


Рисунок 13 – відображення конопки з наведенням та без

Код Css для кнопки при натисканні рис.14

```

.QPushButton:pressed{
line-height: 48px;
color: #ccc;
background-color: #b42f32;
border-color: #1c1c1c #202020 #222;
background-image: -webkit-linear-gradient(top, #a3161a, #b63335
60%, #bf4749);
background-image: -moz-linear-gradient(top, #a3161a, #b63335 60%,
#bf4749);
background-image: -o-linear-gradient(top, #a3161a, #b63335 60%,
#bf4749);
background-image: linear-gradient(to bottom, #a3161a, #b63335
60%, #bf4749);
-webkit-box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.3), 0 1px
rgba(255, 255, 255, 0.09);
box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.3), 0 1px rgba(255,
255, 255, 0.09);
}

```



Рисунок 14 – кнопки при натисканні

### Перетворення Python в автономні виконувані файли

Для перетворення файлів Python було використано PyInstaller який заморожує (пакеми) додатки Python в автономні виконувані файли, під Windows, GNU / Linux, Mac OS X, FreeBSD, Solaris і AIX.

Основні переваги PyInstaller перед аналогічними інструментами полягають у тому, що PyInstaller працює з Python 2.7, 3.5-3.7, він буде менше виконувані файли завдяки прозорому стиску, це повністю мультиплатформовий, і використовуйте підтримку ОС для завантаження динамічних бібліотек, таким чином, забезпечуючи повну сумісність.

## ВИСНОВКИ

В процесі створення кваліфікаційної роботи було проведено порівняння мов програмування для з'ясування мови який підходить за вимогами після було вибрано програму для створення графічної оболонки, був створений програвач з елементами зупинки та продовження, було сторено функцію відключення аудіо та функція списку, створенна зовнішня оболонка , За допомогою сії був створенний зовнішній вигляд програми

Для досягнення мети були виконані наступні пункти:

- крос платформенна програма;
- невелика вага програми;
- програвання різних аудіо форматів;
- створення програвача з можливостями вибору аудіо файлу.

Подальшому буде розроблено:

- систему конвертації аудіо файлів
- можливості створення альбомів
- зміна зовнішнього вигляду
- відтворення візуального вигляду аудіо треків

## Перелік джерел посилання

1. Кульман, Дэйв. "Книга на Python: начало Python, расширенный Python и Python упражнения" URL: [https:// web.archive.org/ web/ 20120623165941/ http://cutter. rexx. com /~dkuhlman /python\\_book \\_01.html](https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html)
2. Peterson, Benjamin (20 Апреля 2020). "Python Insider: Python 2.7.18, последний релиз Python 2" URL: [https://pythoninsider. blogspot.com/2020/04/python-2718-last-release-of-python-2.html](https://pythoninsider.blogspot.com/2020/04/python-2718-last-release-of-python-2.html)
3. Hettinger, Raymond (30 января 2002 года). "PEP 289-генератор выражений" URL: [https://www.python.org / dev/peps/pep-0289/](https://www.python.org/dev/peps/pep-0289/)
4. Peters, Tim (19 августа 2004 года). "PEP 20-Дзен питона" URL: [https://www.python.org/ dev/peps/pep-0020/](https://www.python.org/dev/peps/pep-0020/)
5. "Є гарною мовою Python для початківців програмістів?". URL: [https://docs.python.org/ 3/faq/ general.html#is- python- a-good- language- for- beginning programmers](https://docs.python.org/3/faq/general.html#is-python-a-good-language-for-beginning-programmers)
6. Eby, Phillip J. (7 Декабря 2003 Года). "PEP 333-Python Web Server Gateway Interface v1.0" URL: <https://www.python.org/dev/peps/pep-0333/>
7. Prechelt, Lutz (14 Марта 2000 Года). Емпіричне порівняння C, C++, Java, Perl, Python, Rexx и Tcl URL: [http://page.mi.fu- berlin.de/prechelt/Biblio/jccpprt\\_computer2000.pdf](http://page.mi.fu-berlin.de/prechelt/Biblio/jccpprt_computer2000.pdf)
8. Руководство для разработчиков CSS" URL: [https:// developer. mozilla .org/ en-US/docs/Learn/CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS)
9. "Що таке CSS?" URL: [https:// www.w3.org/ standards/ webdesign/ htmlcss#whatcss](https://www.w3.org/standards/webdesign/htmlcss#whatcss)
10. "Специфікація W3C CSS2. 1 для наборів правил, блоків оголошень і селекторів" URL: [https:// www.w3.org/ TR/ CSS21/ syndata.html#q10](https://www.w3.org/TR/CSS21/syndata.html#q10)
11. "HTML 5. Словник і пов'язані з ним API для HTML і XHTML" URL: [https://html.spec.whatwg.org/multipage/introduction.html#presentational- markup](https://html.spec.whatwg.org/multipage/introduction.html#presentational-markup)

12. Meyer, David (24 Жовтня 2011). "Nokia дає Qt управління з відкритим вихідним кодом" URL: <https://www.zdnet.com/article/nokia-gives-qt-open-source-governance/>
13. "QtDoc 5.1: Всі Модулі" URL: <https://doc.qt.io/qt-5/qtmodules.html>
14. Qt. "Legal-FAQ-Qt" URL: <https://www.qt.io/faq/>