

Projekt Python-19:

Sprawozdanie końcowe

Wstęp

Uwaga: Dokument stanowi załącznik do dokumentacji Specyfikacja_funkcjonalna_Python-19.pdf oraz Specyfikacja_implementacyjna_Python-19.pdf.

W ciągu miesiąca, kiedy powstawał projekt, zachodziły różne dynamiczne zmiany. W związku z tym zgodnie z zapowiedzią zawartą w dokumencie Specyfikacja_implementacyjna_Python-19.pdf, postanowiliśmy stworzyć raport zmian (sprawozdanie końcowe), w którym przedstawimy wszystkie najważniejsze zmiany wprowadzone podczas implementacji powyższego projektu.

Nowa struktura projektu

Tak jak wcześniej, projekt został podzielony na dwa główne pakiety: **GUI** oraz **Backend**.

W pakiecie **GUI** znajdują się następujące moduły:

window.py - moduł zawierający klasę Window z layoutem całego okna

tabs.py - moduł zawierający klasę TabsWidget, która posiada dwie zakładki z dwoma wykresami: odpowiednio zakażeń i ozdrowień.

buttons.py – zrezygnowaliśmy z przycisku UpdateBtn (dane aktualizują się automatycznie), natomiast wprowadziliśmy przycisk UncheckBtn, który odznacza wszystkie zaznaczone kraje. Przycisk ReportBtn pozostał tak jak w założeniu.

slider.py - moduł zawierający klasę DoubleSlider, która zawiera dwa suwaki do wyboru daty początkowej i końcowej. Suwaki po zmianie wartości automatycznie aktualizują wykres, co pozwala na dynamiczne i łatwe w obsłudze "przybliżanie" i "oddalanie" części wykresu, która nas interesuje. Oba suwaki posiadają etykiety, które pokazują nam wybraną datę.

checkboxes.py - moduł zawierający klasę Cbx, która tworzy okienko z etykietą (nazwa kraju), które po naciśnięciu odpowiednio dodaje dany kraj do listy krajów w Singletonie.

countries_list.py - moduł zawierający klasę ListWidget. W klasie tej znajduje się lista obiektów klasy Cbx, pasek do przesuwania listy oraz pole do wyszukiwania (obiekt klasy QLineEdit).

graph.py - moduł zawierający klasę GraphWidget, która pozwala na wyświetlanie wykresu, aktualizowanie go oraz tworzenie pliku .png wykorzystywanego do stworzenia raportu PDF.

W pakiecie **Backend** znajdują się moduły:

config.py - prawdopodobnie największa zmiana wprowadzona w projekcie. Moduł ten zastępuje pierwotnie przewidziany moduł data.py. Moduł config.py zawiera klasę Singleton, która przechowuje listę krajów, które wyświetlamy na wykresie oraz krotkę z datą początkową i końcową zakresu wyświetlania. Atrybuty tej klasy są zmieniane w modułach takich jak buttons.py, slider.py, checkboxes.py oraz wykorzystywane przez moduł graph.py.

updates.py - moduł, który odpowiada za automatyczne pobieranie danych z serwera i zapisywanie ich w postaci plików .csv w lokalizacji programu. Dzięki temu zrezygnowaliśmy z przycisku

UpdateButton, ponieważ dane są aktualizowane bez ingerencji użytkownika. Ten moduł inicjalizuje także początkowy zakres dat naszego Singletonu.

report.py - moduł, który tworzy i zapisuje raport w pliku .pdf

exceptions.py - moduł, który zawiera klasy wyjątków:

- UnableToGenerateReportException - wyjątek, który informuje o braku możliwości wygenerowania raportu, jeżeli użytkownik nie zaznaczył żadnych krajów
- UnableToAddMoreCountriesToGraph - wyjątek, który informuje o limicie 6 krajów wyświetlanych na wykresie.

Z wyjątku o braku możliwości zaktualizowania danych zrezygnowaliśmy, ponieważ moduł requests, z którego korzystamy w module updates.py zawiera własny wyjątek, do obsługi braku połączenia z internetem. Zostało natomiast dodane okienko z wiadomością, która informuje nas o wystąpieniu tego wyjątku. W tym przypadku program nie uruchomi się i jedynie pojawi się okienko z błędem.

UWAGA: w załączniku na końcu tego dokumentu zamieszczony został zaktualizowany diagram klas.

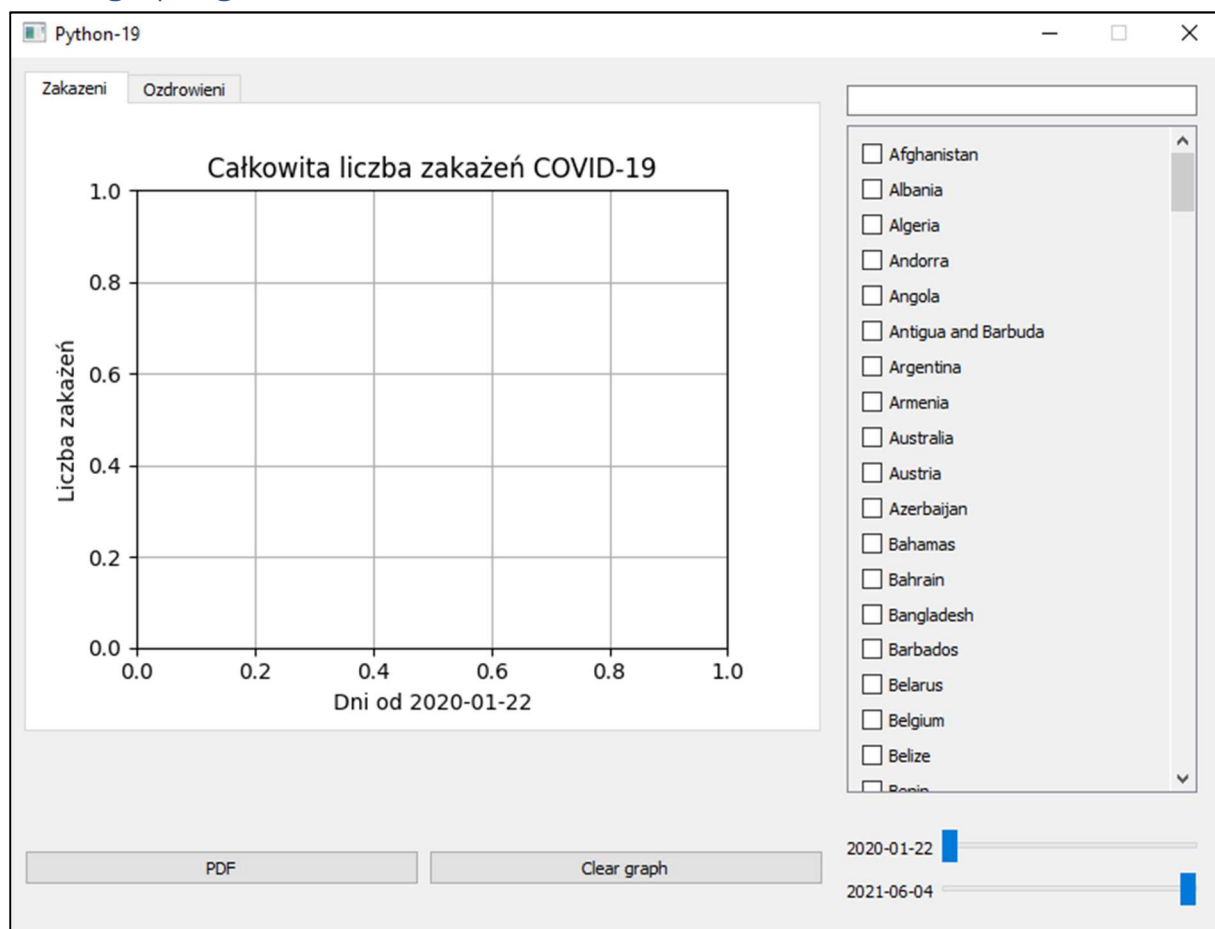
Wykorzystane wzorce projektowe

W naszym projekcie wykorzystaliśmy kreatywny wzorzec projektowy – Singleton. Postanowiliśmy zaimplementować go jako alternatywę dla używania zmiennych globalnych.

Klasa Singleton pozwoliła nam na stworzenie jednego (**i tylko jednego**) obiektu danej klasy. Za pomocą metody Singleton.get_instance() mogliśmy przywołać obiekt tej klasy w każdym miejscu naszego projektu. Tworząc zmienną lokalną data = Singleton.get_instance() otrzymywaliśmy dokładnie ten sam obiekt, o tym samym adresie w pamięci niezależnie od tego gdzie i który raz go przywoływaliśmy.

Dzięki temu z łatwością można było modyfikować atrybuty Singletona w każdej klasie (zaznaczając okienka w klasie Cbx, przesuwając suwaki w klasie DoubleSlider, albo czyszcząc wykres w klasie UncheckBtn). Przywołanie Singletona pozwalało nam później wykorzystać bieżące wartości w klasie GraphWidget do aktualizowania wykresu.

Obsługa programu

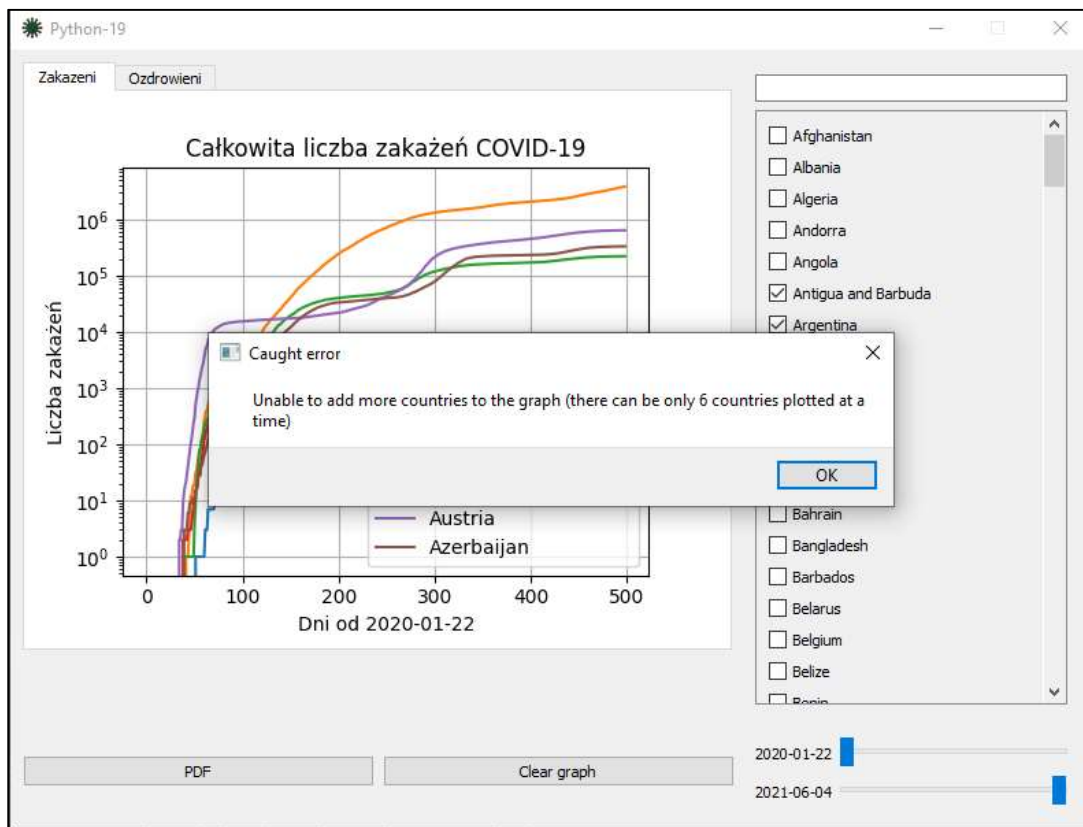


Ekran początkowy naszego programu przedstawiony jest powyżej. W lewym górnym rogu znajdują się dwie zakładki, na każdej zakładce widnieje inny wykres. Oba wykresy aktualizują się za każdym razem gdy zaznaczymy/odznaczmy kraj lub zmienimy zakres dat.

Po prawej stronie widzimy listę krajów posortowaną alfabetycznie. Każdy kraj ma swoje okienko, po którego zaznaczeniu dane dotyczące tego kraju wyświetlane zostają na wykresach. W celu łatwiejszego poruszania się po liście można skorzystać z paska scroll lub/oraz opcji wyszukiwania wpisując część nazwy danego kraju. Po wpisaniu części nazwy lista filtruje się i pozostają jedynie kraje zgodne z wpisaną frazą.

Poniżej listy znajdują się dwa suwaki. Za pomocą suwaka na górze zmieniamy datę początkową, za pomocą suwaka na dole zmieniamy datę końcową. Domyślne ustawienie suwaków obejmuje cały zakres danych.

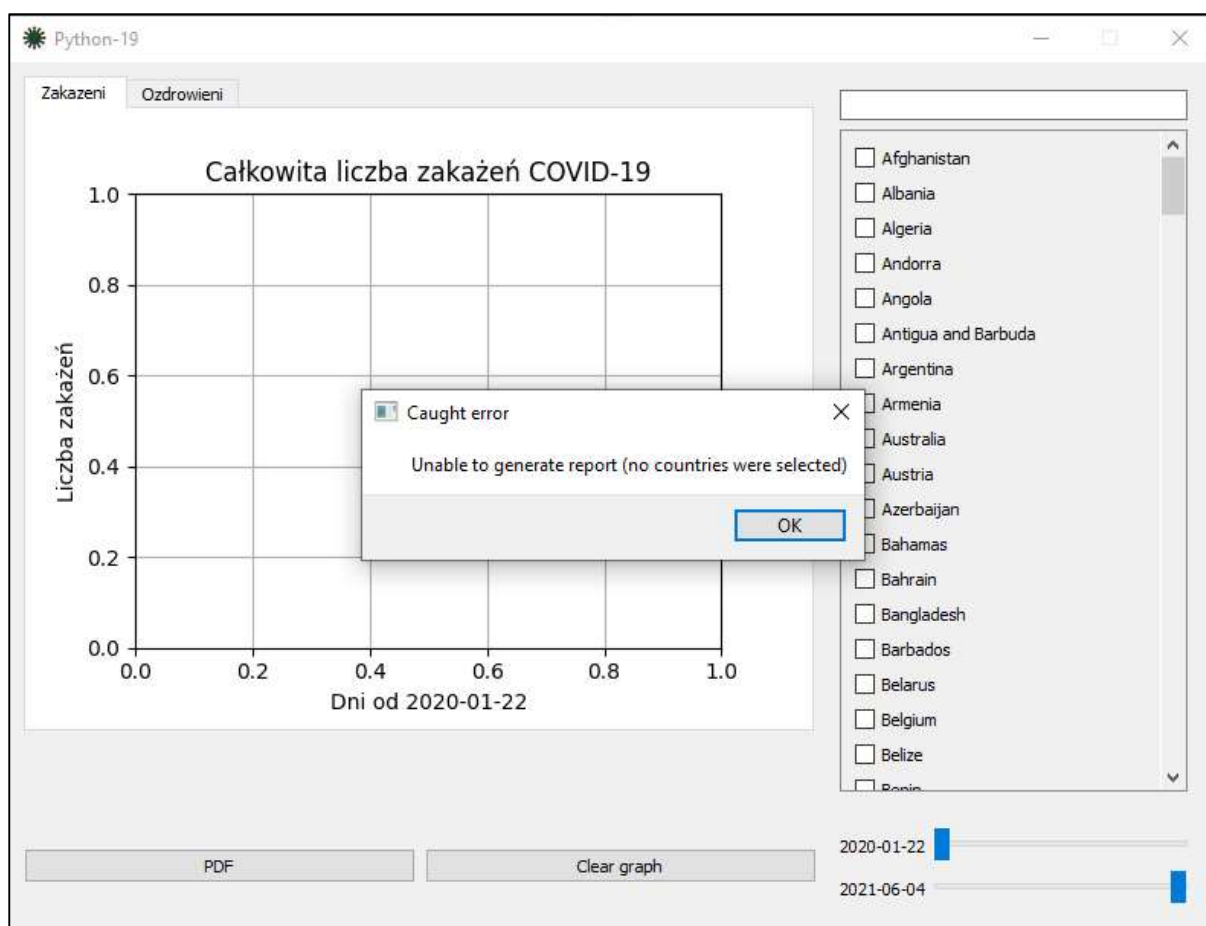
UWAGA! Można wybrać maksymalnie 6 krajów. Jeżeli spróbujemy dodać więcej, program poinformuje nas o błędzie:



W lewym dolnym rogu znajdują się dwa przyciski:

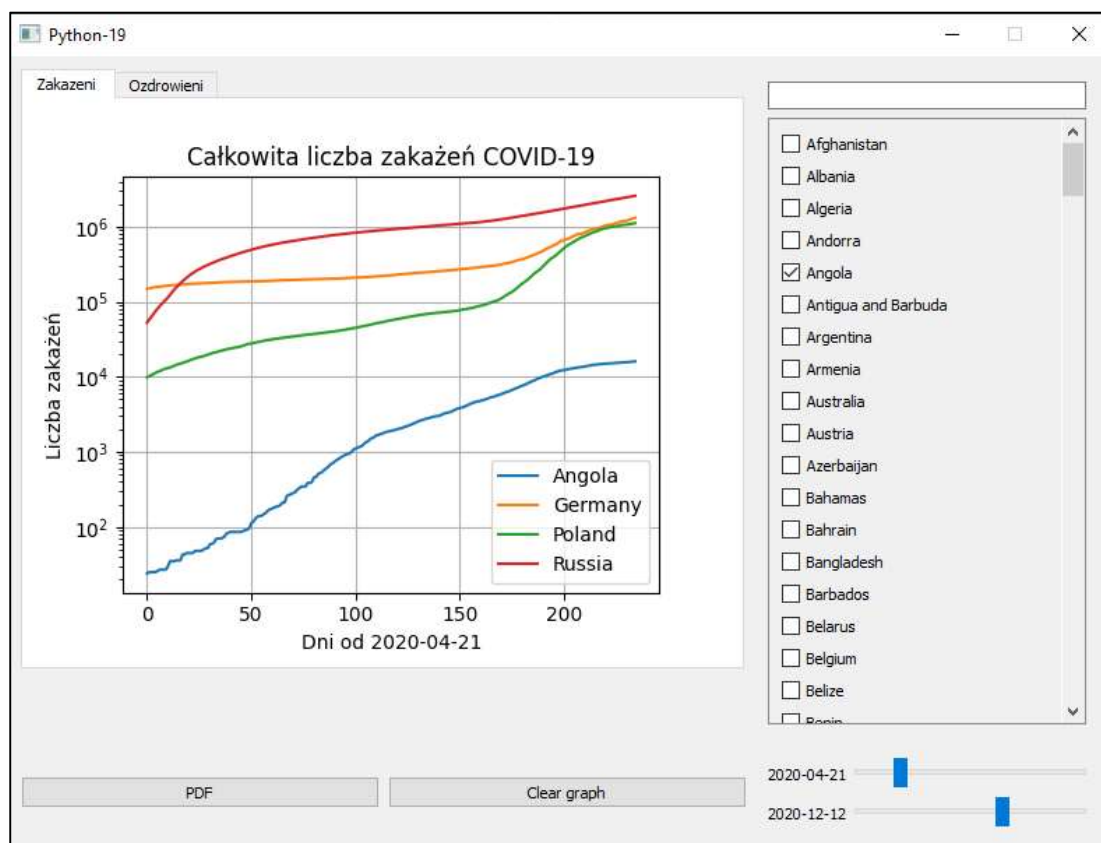
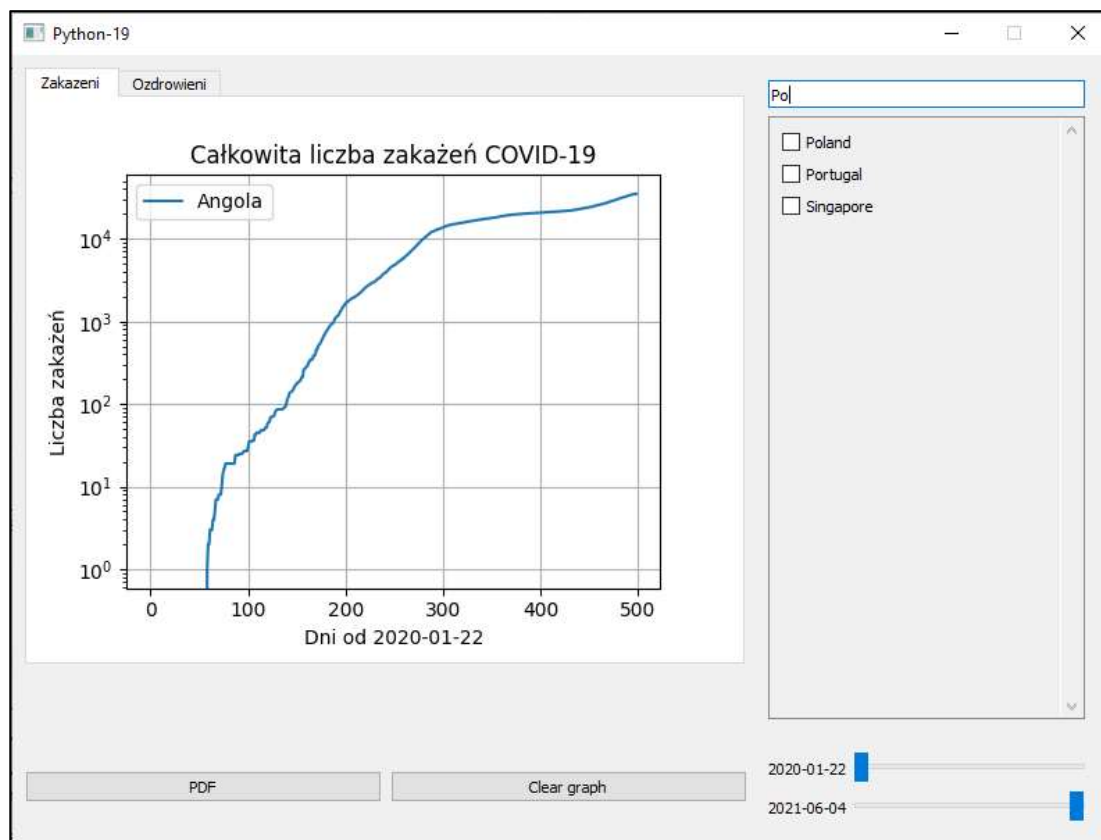
- PDF – jest to przycisk, po którego wciśnięciu otwiera się okienko do wyboru lokalizacji zapisu. Po naciśnięciu przycisku „Zapisz” zostanie utworzony raport PDF z dwoma wykresami wybranych krajów w wybranym zakresie dat.

UWAGA! Jeżeli będziemy chcieli stworzyć raport z pustego wykresu (żadne kraje nie będą zaznaczone), to program poinformuje nas o błędzie:



- Clear graph – po wciśnięciu tego przycisku zostaną odznaczone wszystkie kraje, a wykresy zostaną wyczyszczone.

Przykładowe zrzuty ekranu z działania programu:



Python-19

Zakazeni Ozdrowieni

Całkowita liczba ozdrowień COVID-19

Liczba ozdrowień

Dni od 2020-01-22

PDF Clear graph

2020-01-22 2020-12-28

Save PDF report

← → ↕ << Nowy (D:) > Studia > Projekty > 2021L > Python-19 🔍 Przeszukaj: Python-19

Organizuj Nowy folder

Nazwa	Data modyfikacji	Typ	Rozmiar
.git	05.06.2021 13:45	Folder plików	
.idea	05.06.2021 12:43	Folder plików	
Backend	05.06.2021 12:13	Folder plików	
Documentation	05.06.2021 12:56	Folder plików	
GUI	05.06.2021 13:43	Folder plików	

Nazwa pliku: test

Zapisz jako typ: PDF Files (*.pdf)

Ukryj foldery Zapisz Anuluj

Po naciśnięciu przycisku „Zapisz” powstanie plik test.pdf w wybranym folderze, który będzie zawierał wykres zakażeń z zakładki „Zakazeni” oraz wykres ozdrowień z zakładki widocznej wyżej.

