**Dokumentacja do projektu Configurator PC**

**Artsiom Dziaineka**

**Problem który rozwiązuje ten program**

Program ten rozwiązuje problem sortowania komponentów. Główną funkcją programu jest wybór odpowiednich komponentów. Zaimplementowano funkcję dopasowania płyty głównej do procesora, ponieważ mają one różne gniazda, a także zasilacze są sortowane w zależności od TDP wszystkich komponentów.

**1. Kompatybilność gniazda procesora i płyty głównej:**

Jednym z głównych zadań konfiguratora jest zapewnienie kompatybilności pomiędzy wybranym procesorem i płytą główną. Gniazdo procesora określa, jaki typ procesora można zainstalować na płycie głównej. Wybór płyty głównej z odpowiednim gniazdem pozwala użytkownikom uniknąć sytuacji, w której wybrany procesor nie pasuje do płyty głównej.

1. **Uwzględnienie rozpraszania ciepła:**

(TDP) procesora i zasilacza: Procesory i karty graficzne generują ciepło podczas pracy, znane jako moc cieplna (TDP). Odpowiednie chłodzenie i zasilanie odgrywają kluczową rolę w stabilności i długowieczności komponentów komputera. Konfigurator bierze pod uwagę całkowite TDP wybranych komponentów i sugeruje zasilacze, które mogą zapewnić wystarczającą moc do uruchomienia wszystkich komponentów systemu bez przeciążenia lub przegrzania.

**3. Uproszczenie procesu montażu komputera**: Wybór kompatybilnych komponentów może być trudnym zadaniem dla niedoświadczonych użytkowników. Konfigurator upraszcza ten proces, sugerując tylko kompatybilne opcje płyt głównych i zasilaczy w oparciu o wybrane parametry, takie jak gniazdo procesora i wymagania dotyczące zasilania.

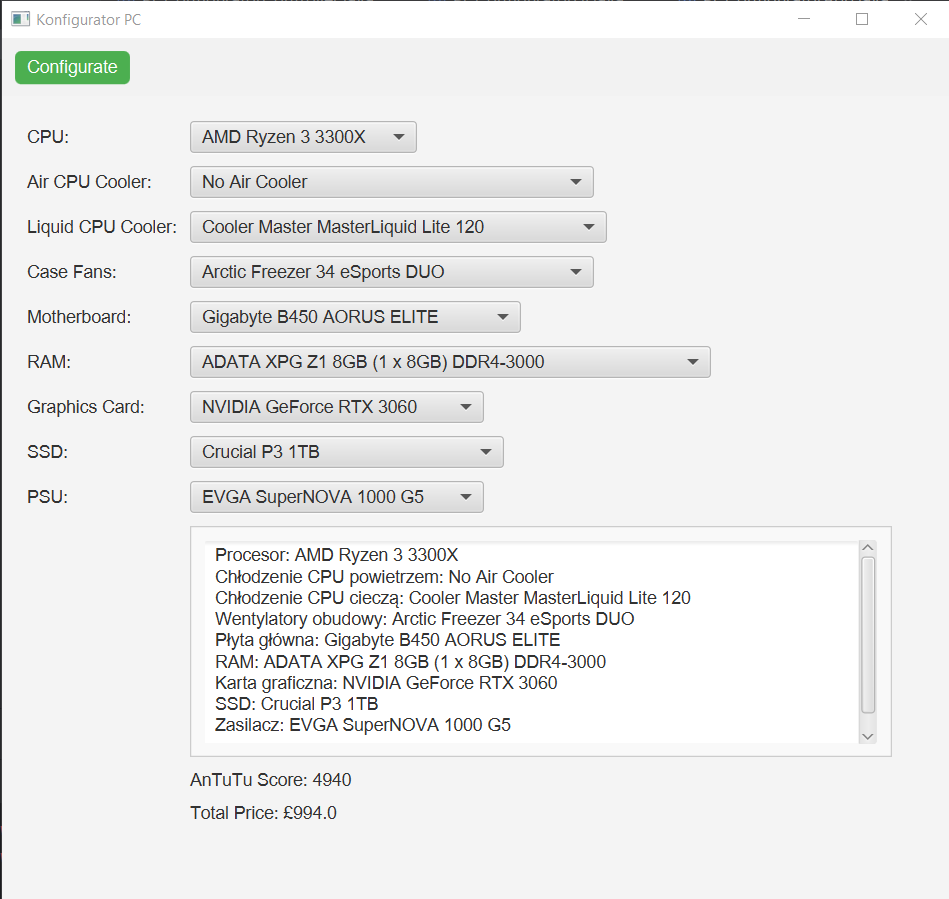
**4. Oszczędność czasu i zasobów**: Zamiast ręcznie wyszukiwać kompatybilne komponenty lub kupować je metodą prób i błędów, użytkownik może skorzystać z konfiguratora, aby szybko i dokładnie wybrać odpowiednie komponenty, oszczędzając czas i pieniądze.

Tak więc konfigurator PC, który dopasowuje płytę główną do konkretnego gniazda i sortuje zasilacze na podstawie TDP, rozwiązuje problemy kompatybilności, wydajności i wygody podczas składania systemów komputerowych.

Jeśli uwzględni się w tym programie podzespoły z ostatnich 5 lat, będzie to niezwykle przydatna aplikacja do celów komercyjnych, a być może nawet do użytku osobistego

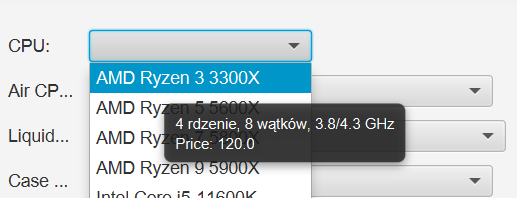
**Opis sposobu korzystania z programu w formie krótkiej instrukcji dla użytkownika:**

Cały program wygląda bardzo prosto i jest również łatwy w użyciu. Mamy komórki, w których wybieramy dostępne komponenty



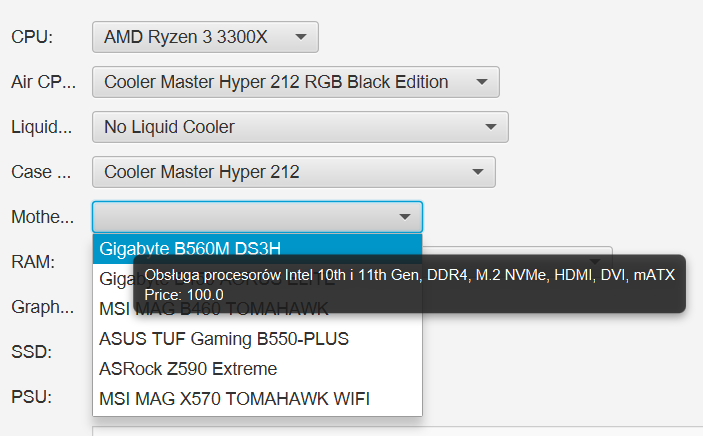
Program, po najechaniu kursorem na dany komponent, wyświetla opis parametrów technicznych ważnych dla użytkownika oraz cenę, co znacznie ułatwia wybór komponentów w oparciu o osobiste preferencje i budżet .

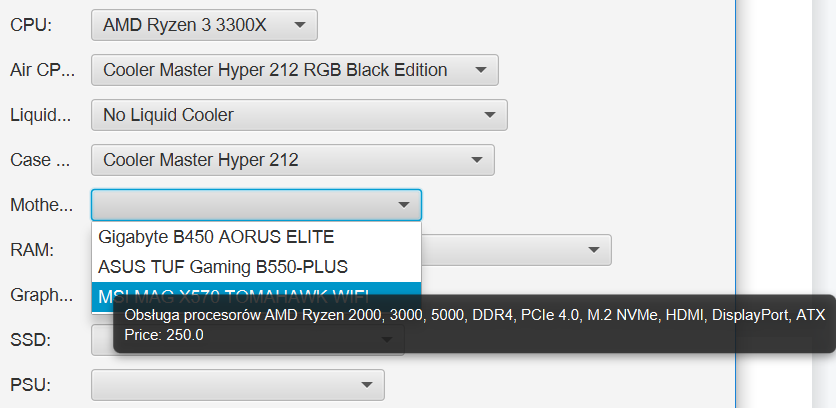
Ta funkcja działa dla wszystkich komponentów.



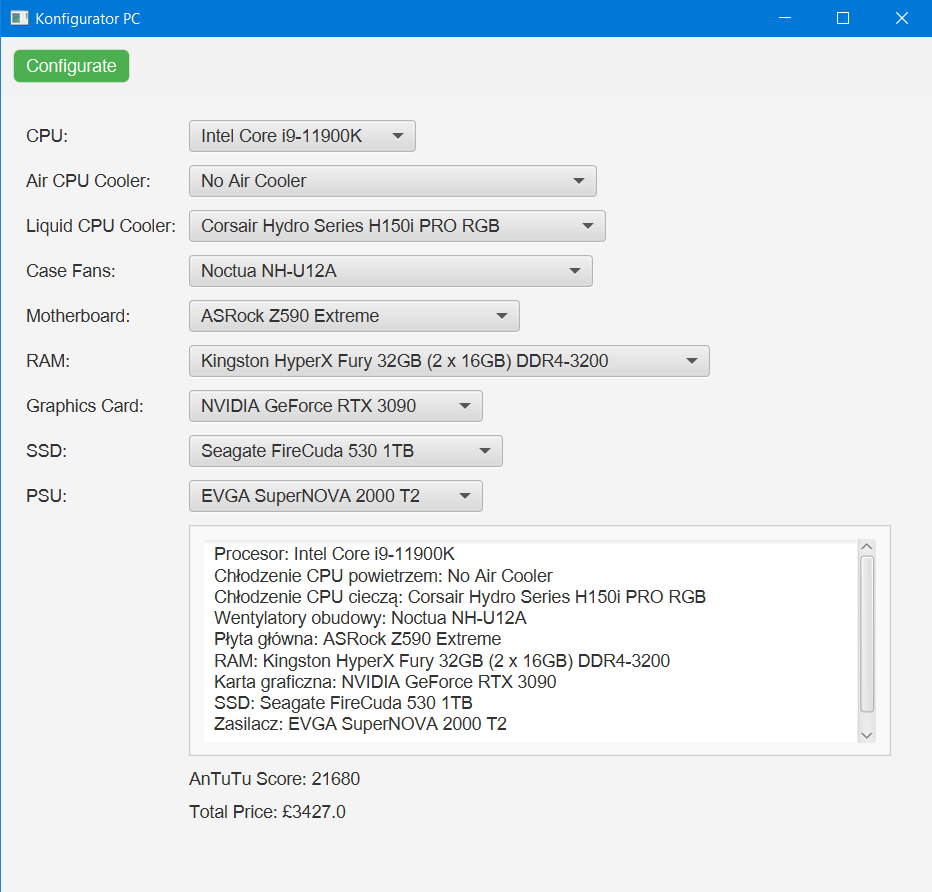
**niestety nie bez bagów :(**

*Przy wyborze płyty głównej, wszystkie są początkowo dostępne, gniazdo nie jest brane pod uwagę. Musisz najpierw kliknąć jedną z płyt, a następnie pojawią się posortowane płyty. Nie rozwiązałem tego błędu. Niestety. Spędziłem około 40-50 godzin nad projektem i już się w nim gubię, ale myślę, że najważniejsze jest to, że działa : )*





Po zapoznaniu się ze wszystkimi podzespołami i złożeniem należy kliknąć na przycisk Configure, który uruchomi funkcję wypisania całego złożenia komputera, tzw. "AnTuTu" - oceny mocy obliczeniowej komputera (nie zaimplementowałem tego tak jak powinno działać, ale pomysł działa, w przyszłości można by zrobić zapytania i działałoby jak trzeba) oraz bezpośrednio ceny złożenia.



**Dodatkowy opis każdego klasa i metod**

**Klasa PCComponent**

**Opis**

PCComponent jest klasą bazową dla wszystkich komponentów PC. Zawiera wspólne właściwości i metody, które będą używane przez wszystkie komponenty.

**Właściwości**

private String name: Nazwa komponentu.

private double price: Cena komponentu.

private String description: Opis komponentu.

**Metody**

Public String getName(): Zwraca nazwę składnika.

public double getPrice(): Zwraca cenę składnika.

public String getDescription(): Zwraca opis komponentu.

public String toString(): Zwraca reprezentację łańcuchową komponentu w formacie "name (£price)".

**Klasa CPU**

**Opis**

CPU jest klasą reprezentującą procesor, odziedziczoną po PCComponent.

**Właściwości**

private String socket: Gniazdo procesora.

private int TDP: Moc obliczeniowa procesora.

**Konstruktor** public CPU(String name, double price, String description, String socket, int TDP)

name: Nazwa procesora.

price: Cena procesora.

description: Opis procesora.

socket: Gniazdo procesora.

TDP: Pobór mocy procesora.

**Metody**

public String getSocket(): Zwraca gniazdo procesora.

public int getTDP(): Zwraca pobór mocy procesora.

**Klasa GPU**

**Opis**

GPU jest klasą reprezentującą kartę graficzną, odziedziczoną po PCComponent.

**Właściwości**

private int TDP: Pobór mocy karty graficznej.

**Konstruktor** public GPU(String name, double price, String description, int TDP)

name: Nazwa karty graficznej.

price: Cena karty graficznej.

description: Opis karty graficznej.

TDP: Pobór mocy karty graficznej.

**Metody**

public int getTDP(): Zwraca pobór mocy karty graficznej.

**Klasa płyty głównej**

**Opis**

Motherboard jest klasą reprezentującą płytę główną, odziedziczoną po PCComponent.

**Właściwości**

Private String socket: gniazdo płyty głównej.

Konstruktor public Motherboard(String name, double price, String description, String socket)

name: Nazwa płyty głównej.

price: Cena płyty głównej.

description: Opis płyty głównej.

socket: Gniazdo płyty głównej.

**Metody**

public String getSocket(): Zwraca gniazdo płyty głównej.

**Klasa PCConfiguratorApp**

**Opis**

PCConfiguratorApp jest klasą reprezentującą główną aplikację konfiguratora PC. Dziedziczy po klasie Application i jest używana do uruchamiania aplikacji JavaFX.

**Właściwości**

private PCConfiguratorController controller: Kontroler do sterowania aplikacją.

**Metody**

public void start(Stage primaryStage): Metoda uruchamiająca główne okno aplikacji.

public static void main(String[] args): Punkt wejścia do aplikacji.

**Klasa PCConfiguratorUI**

**Opis**

PCConfiguratorUI jest klasą reprezentującą interfejs użytkownika dla konfiguratora PC.

**Właściwości**

private BorderPane rootPane: Główny panel interfejsu.

private GridPane grid: Siatka do umieszczania elementów interfejsu.

private Label totalPriceLabel: Etykieta wyświetlająca całkowity koszt konfiguracji.

private TextArea configurationTextArea: Pole tekstowe do wyświetlania bieżącej konfiguracji.

private Label antutuScoreLabel: Etykieta wyświetlająca wynik AnTuTu.

private Tooltip descriptionTooltip: Tooltip do wyświetlania opisów komponentów.

private PCConfiguratorController controller: Kontroler do zarządzania interfejsem.

**Konstruktor** public PCConfiguratorUI(PCConfiguratorController controller)

controller: Kontroler do zarządzania interfejsem.

**Metody**

**public BorderPane getRootPane():** Zwraca panel główny interfejsu.

**private ComboBox<PCComponent> createComboBox(PCComponent[] components):** Tworzy okno kombi do wyboru komponentów.

**private <T extends PCComponent> void addDescriptionToolTip(ComboBox<T> comboBox):** Dodaje podpowiedzi opisu dla komponentów w polach kombi.

**private void filterMotherboards(ComboBox<Motherboard> motherboardComboBox, ObservableList<Motherboard> allMotherboards, String socket):** Filtruje listę płyt głównych według gniazda.

**Klasa PCConfiguratorController**

**Opis**

PCConfiguratorController jest klasą reprezentującą kontroler do sterowania logiką konfiguratora PC.

**Metody**

**public void setComboBoxes(ComboBox<PCComponent> airCpuCoolerComboBox, ComboBox<PCComponent> liquidCpuCoolerComboBox, ComboBox<PCComponent> caseFansComboBox, ComboBox<PCComponent> ramComboBox, ComboBox<PCComponent> graphicsCardComboBox, ComboBox<PSU> psuComboBox, ComboBox<PCComponent> ssdComboBox, ComboBox<Motherboard> motherboardComboBox, ComboBox<CPU> cpuComboBox)**: Ustawia pola kombi do wyboru komponentów.

**public void refreshPSUList():** Odświeża listę zasilaczy na podstawie wybranych komponentów.

Od siebie mogbym powiedzieć, że dla mnie projekt jest jednym z powodów szczęścia, chociaż ma bag.