

# Projekt – Zaawansowane Programowanie Obiektowe i Funkcyjne 2025Z – konspekt

---

---

## Podstawowe informacje

### Tytuł projektu –

Prognozowanie wyników inwestycyjnych z kwantyfikacją ryzyka

### Członkowie zespołu –

- Adam Bagiński (338978)
  - Aleksandra Dmitruk (338981)
  - Barbara Gawlik (338987)
- 

## Ogólna charakterystyka, cel projektu i jego opis biznesowy

- **Ogólna charakterystyka** – Aplikacja desktopowa lub webowa o nazwie *RiskFolio*, służąca do budowania wirtualnego portfela aktywów i symulowania jego przyszłej wartości w oparciu o dane historyczne i prognozowanie metodą Monte Carlo.
  - **Cel projektu** – Stworzenie narzędzia, które pozwoli użytkownikowi nie tylko sprawdzić historyczne stopy zwrotu, ale przede wszystkim odpowiedzieć na pytania takie jak:
    - *Czy jeśli planuję za rok sprzedać akcje danej firmy, to na tym stracię?*
    - *Kiedy powiniem sprzedać, a kiedy kupować dany instrument finansowy?*
    - *Ille mogę zarobić w pesymistycznym i optymistycznym scenariuszu?*
    - *Ille maksymalnie i minimalnie mogę stracić na tej inwestycji?*
  - **Opis biznesowy** – Początkujący inwestorzy często patrzą tylko na średni zysk, ignorując ryzyko. Aplikacja dostarcza więc kwantyfikację tego ryzyka, pomagając w podjęciu bardziej świadomej decyzji inwestycyjnej i zarządzaniu oczekiwaniami.
- 

## Wymagania funkcjonalne (m.in. założenia i funkcje)

1. **Zarządzanie portfelem** – użytkownik
  - a. dodaje aktywa (np. AAPL, MSFT, TSLA),
  - b. określa skład portfela (np. 40% AAPL, 30% MSFT, 30% TSLA),
  - c. podaje kapitał początkowy (np. 100 000 PLN),
  - d. podaje horyzont prognozy (np. 365 dni).

2. **Weryfikacja danych** – system zwróci błąd w przypadku wprowadzenia nieprawidłowych danych przez użytkownika – np. ujemny horyzont czasowy lub nieistniejący / nieobsługiwany ticker.
  3. **Pobieranie danych** – aplikacja automatycznie pobiera historyczne notowania (np. z ostatnich 5 lat) z zewnętrznego API (np. Yahoo Finance).
  4. **Analiza statystyczna** – obliczanie
    - a. podstawowych metryk dla każdego aktywa (np. średni zwrot, odchylenie standardowe),
    - b. korelacji między aktywami (np. czy jak Bitcoin spada, to złoto rośnie?)
  5. **Symulacja Monte Carlo** – wygenerowanie 10,000 możliwych scenariuszy (wyniku inwestycji) dla portfela na podstawie historycznych danych – system musi wyświetlić tabelę z wynikami symulacji np. średni prognozowany zwrot i wartość portfela.
  6. **Kwantyfikacja ryzyka** – obliczanie VaR i CVaR (np. z prawdopodobieństwem 95% nie stracisz więcej niż 1000 PLN w ciągu jednego miesiąca).
  7. **Wizualizacja** – wykres wartości portfela na przestrzeni czasu.
- 

## Wymagania niefunkcjonalne

1. **Programowanie funkcyjne** – wykorzystanie Java Streams API do przetwarzania danych finansowych (np. map, reduce, filter) oraz interfejsów funkcyjnych (np. Supplier, Function) do strategii obliczania ryzyka.
  2. **Programowanie obiektowe** – wykorzystanie polimorfizmu (np. różne typy aktywów dziedziczące po klasie abstrakcyjnej Asset).
  3. **Wydajność** – wykonywanie pobierania danych i obliczania symulacji musi zostać wykonane w czasie maksymalnie 10 sekund przy pomocy np. ParallelStream, aby nie blokować interfejsu użytkownika.
  4. **Czysty i czytelny kod** – docstringi opisujące każdy plik z kodem, plik README, obsługa wyjątków i konwencja nazewnictwa funkcji, obiektów oraz zmiennych.
  5. **Intuicyjność aplikacji** – większość użytkowników powinna być w stanie poprawnie skonfigurować swój portfel i uruchomić symulację w czasie krótszym niż 2 minuty. Interfejs użytkownika dostępny jest w języku polskim.
  6. **Bezpieczeństwo** – aplikacja nie przechowuje danych użytkowników (np. skład portfela) – dane są automatycznie kasowane po zamknięciu sesji
  7. **Niezawodność** – model obliczeniowy poprawnie obsługuje aktywa (*do ustalenia jak*), dla których nie ma wystarczającej ilości danych i poinformuje o tym użytkownika.
- 

## Planowane technologie i biblioteki

- **Język + IDE** – Java 21 + IntelliJ IDEA 2024.
- **GUI** – JavaFX.
- **Dane** – YahooFinanceAPI.
- **Obliczenia** – Apache Commons Math / implementacja własna.
- **Zarządzanie** – BitBucket, Google Spreadsheets, Google Drive.