

Dane pobrane ze zbioru fashion mnist po 20 zdjęć tshirt, dress, pullover.

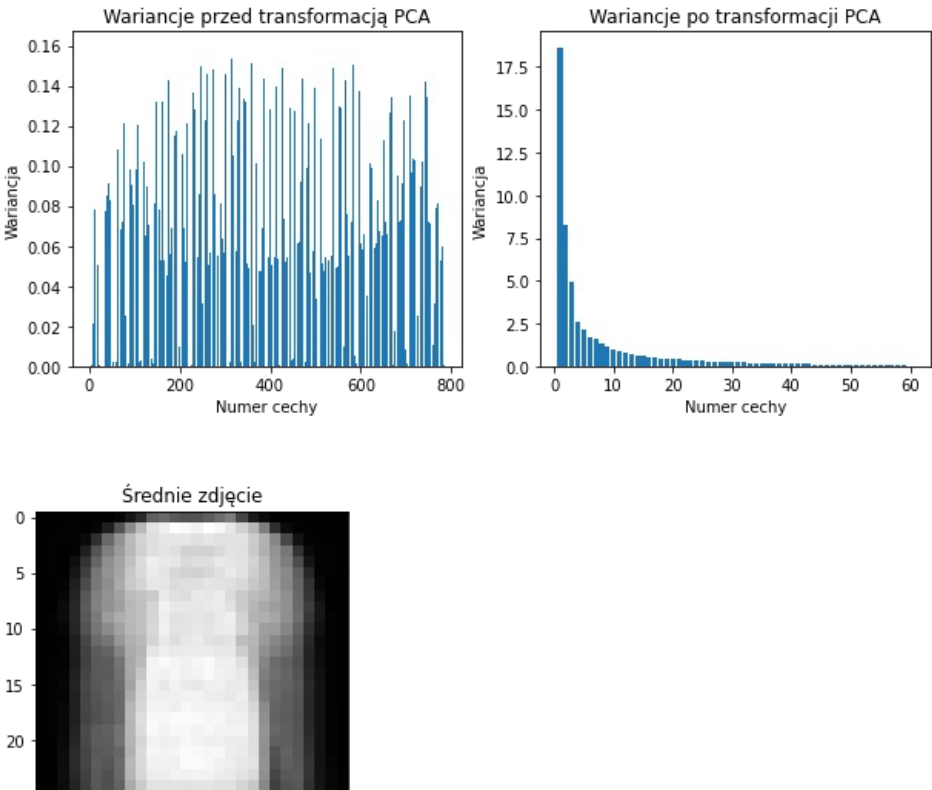


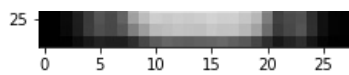
Wymiary macierzy kowariancji przed PCA: (784, 784)
Wymiary wektora wariancji przed PCA: (784,)

```
Out[93]: PCA
PCA()
```

Wymiary macierzy kowariancji po PCA: (60, 784)
Wymiary wektora wariancji po PCA: (60,)

1. Macierz cowariancji przed to symetrycznamacierz 784x784 każda cecha to jeden piksel.





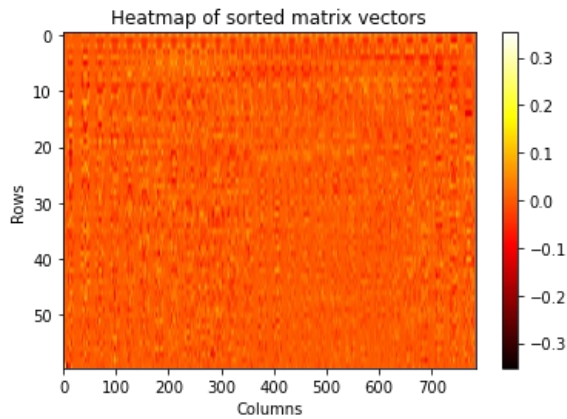
Widac że średnie zdjęcie ma zarys elementu garderoby.

In [132]

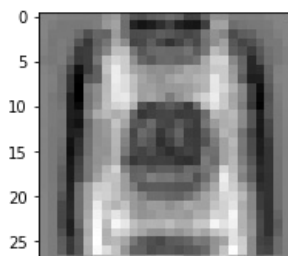
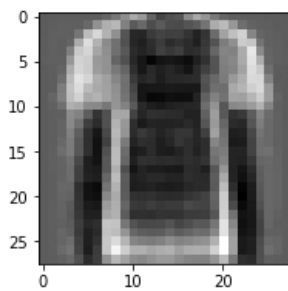
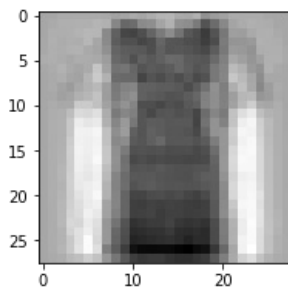
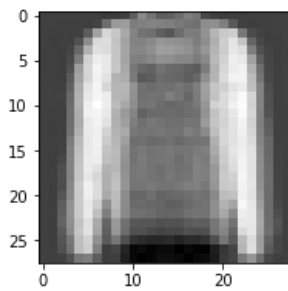
```
# Posortowanie wektorów bazowych według powiązanej wariancji
sorted_idx = np.argsort(explained_variance)[::-1] # Indeksy posortowane malejąco
principal_components_sorted = principal_components[sorted_idx]

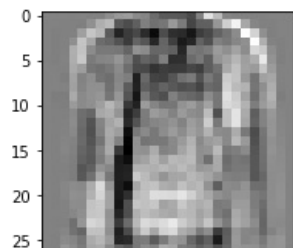
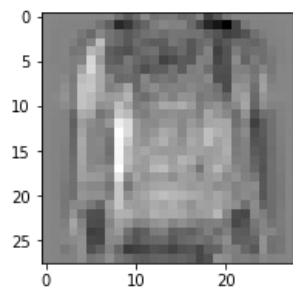
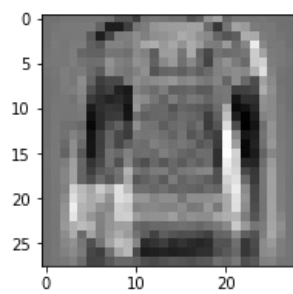
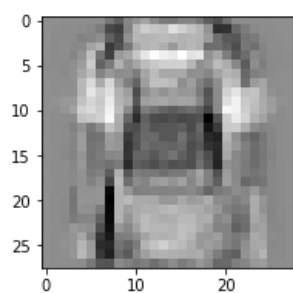
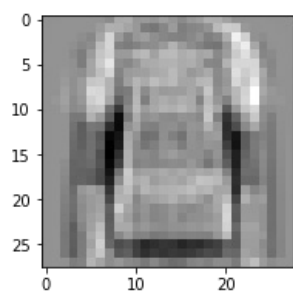
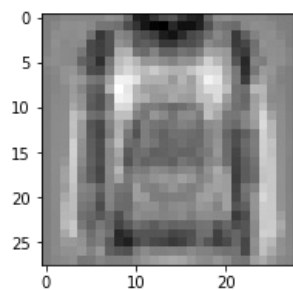
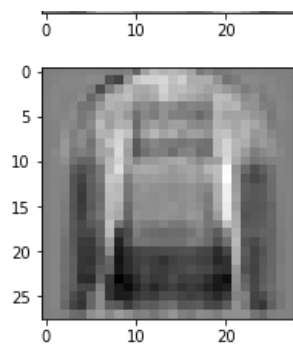
print(principal_components_sorted.shape)
```

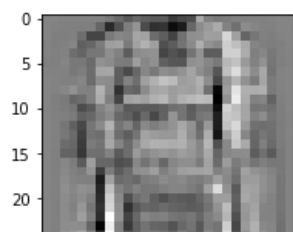
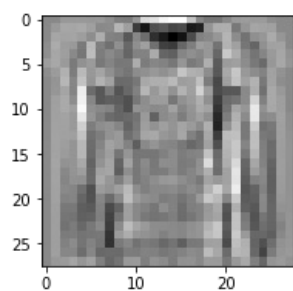
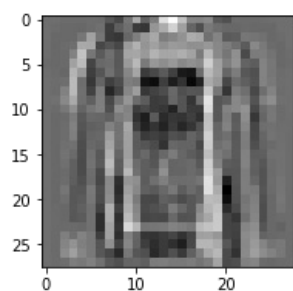
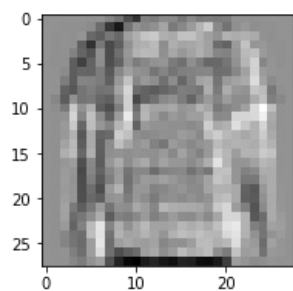
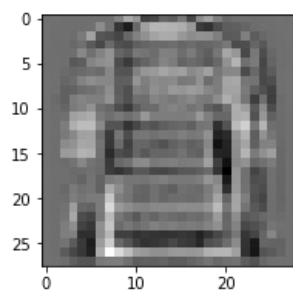
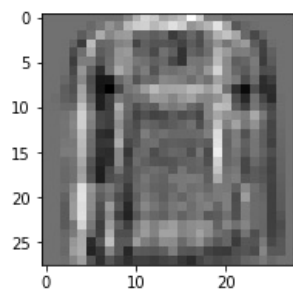
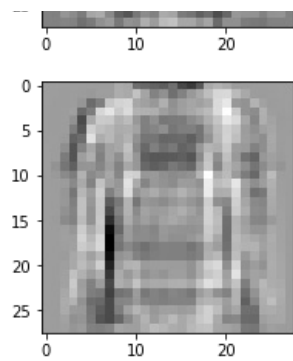
(60, 784)

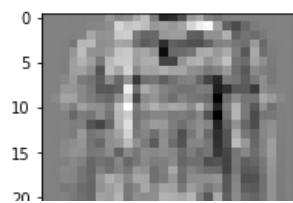
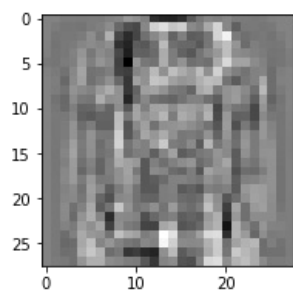
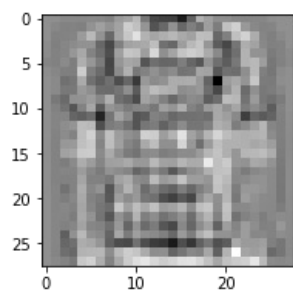
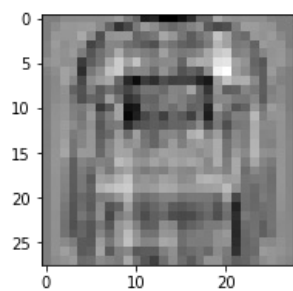
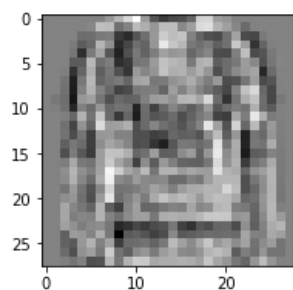
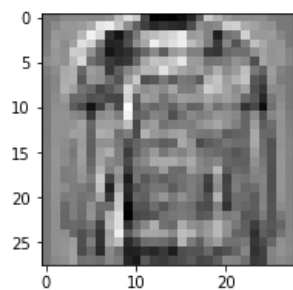
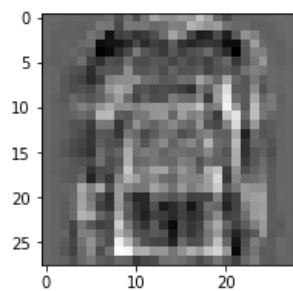
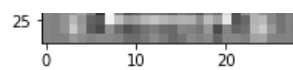


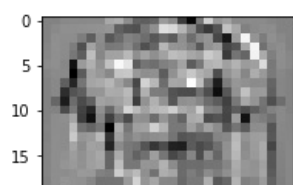
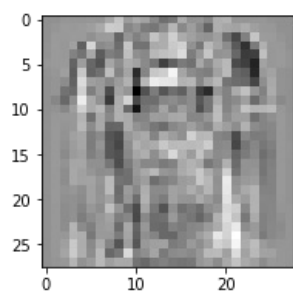
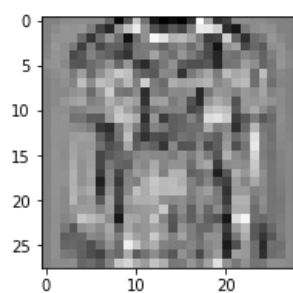
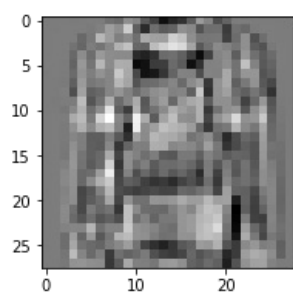
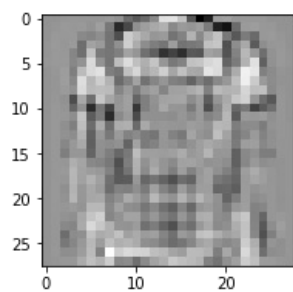
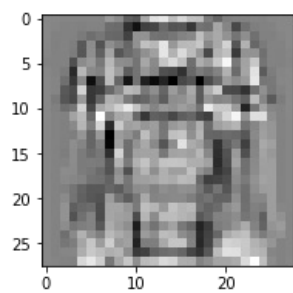
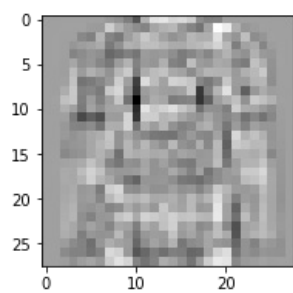
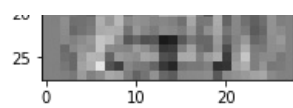
Obrazy powstałe z nowo znalezionej bazy:

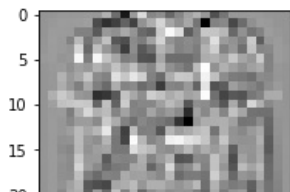
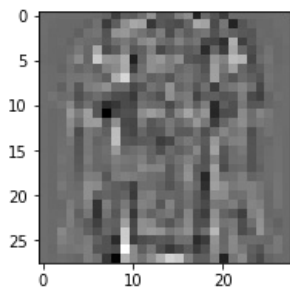
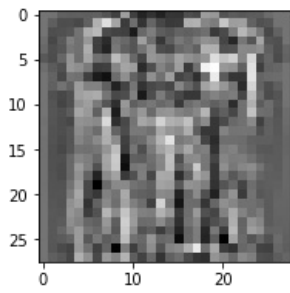
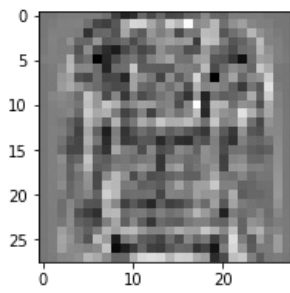
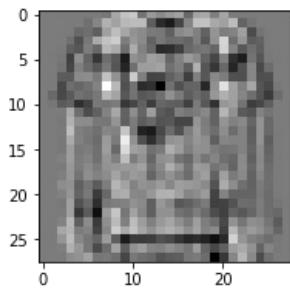
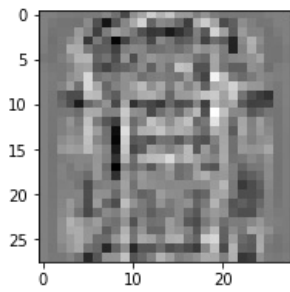
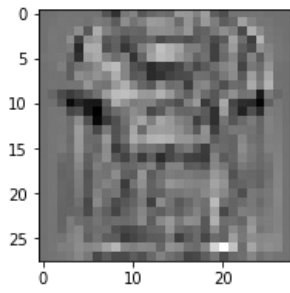
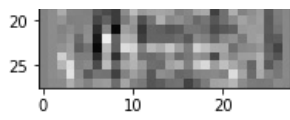


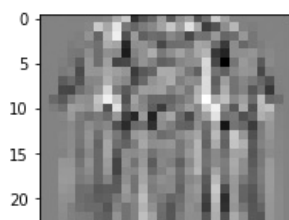
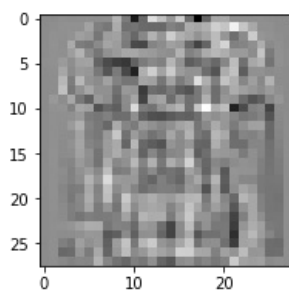
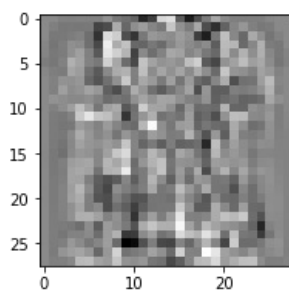
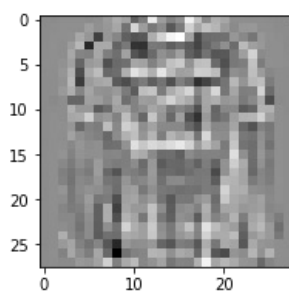
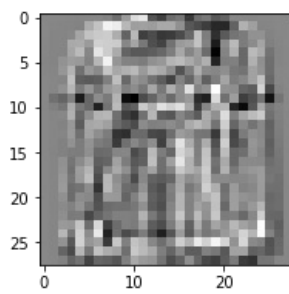
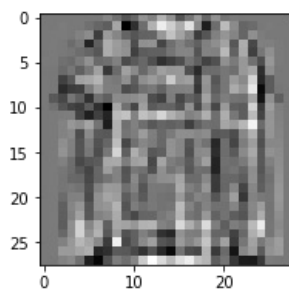
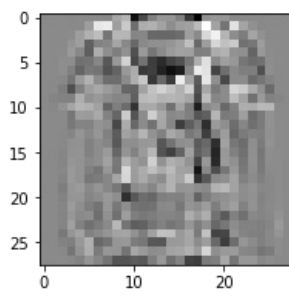
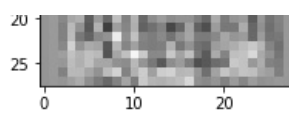


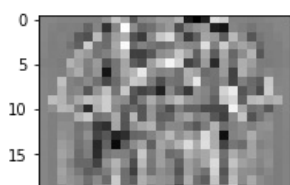
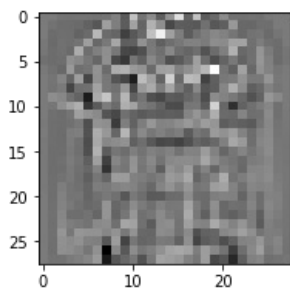
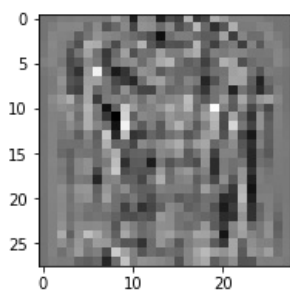
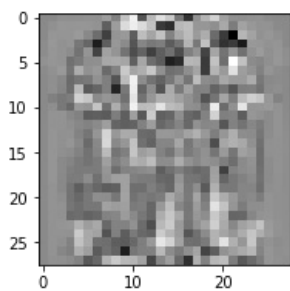
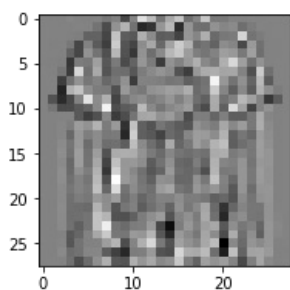
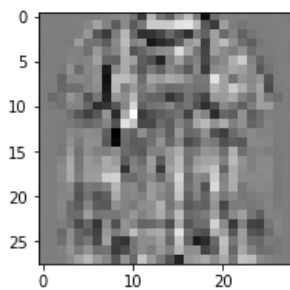
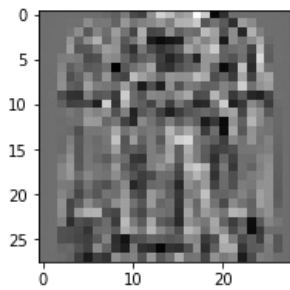
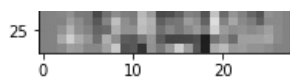


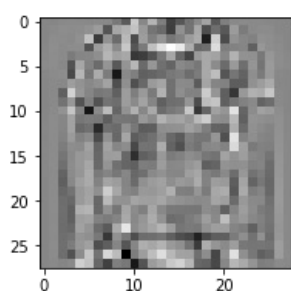
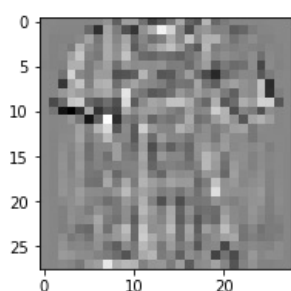
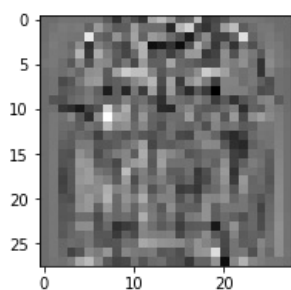
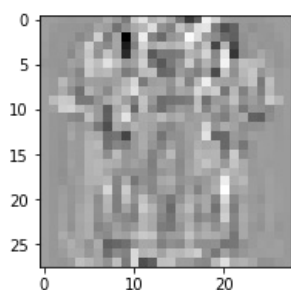
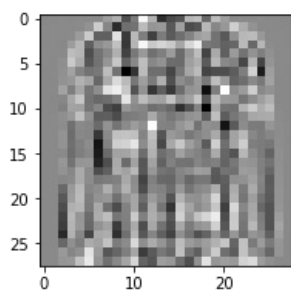
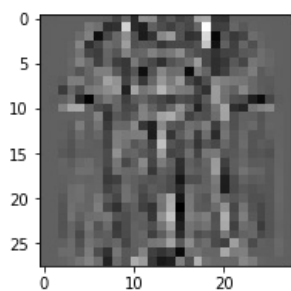
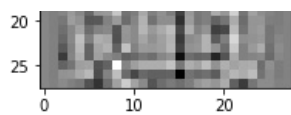


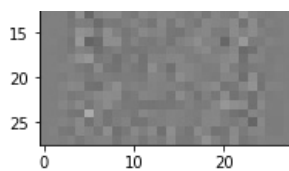












Znacznie mniejsza jakość obrazków w stosunku do pierwotnych.

In [155..

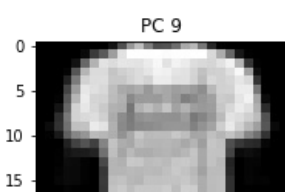
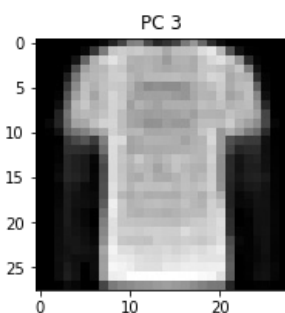
```
n_components_list = [3, 9, 27]
X_centered = centered
X_mean = mean

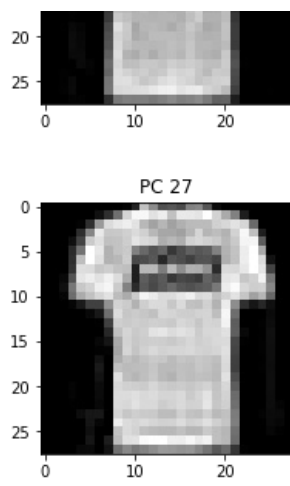
for n_components in n_components_list:
    # Wyzerowanie wartości cech poza wybraną garstką
    X_reduced = np.dot(X_centered, pca.components_[:n_components].T) # Konwersja do nowej bazy
    X_reduced[:, n_components:] = 0 # Wyzerowanie zbędnych cech

    # Przywrócenie do oryginalnej bazy
    X_restored = np.dot(X_reduced, pca.components_[:n_components]) # Konwersja do oryginalnej bazy
    X_restored += X_mean # Dodanie średniego wektora

    # Przekształcenie wektorów cech na obrazy
    X_restored = np.clip(X_restored, 0, 1) # Ograniczenie wartości pikseli do przedziału [0, 1]
    X_restored = (X_restored * 255).astype(np.uint8) # Przeskalowanie wartości pikseli do zakresu [0, 255]
    X_restored = X_restored.reshape(-1, 28, 28) # Przekształcenie wektorów cech na obrazy

    # Wyświetlenie "odchudzonych" fotografii
    plt.figure(figsize=(3,3))
    plt.imshow(X_restored[0], cmap='gray') # Wyświetlenie obrazu
    plt.title(f'PC {n_components}') # Numer głównej składowej
    plt.show()
```





Powyżej przykład tego samego zdjęcia dla różnej liczby wymiarów. Wraz z rosnącą liczbą komponentów rośnie jakość zdjęcia.

Wizualizacja zbioru na płaszczyźnie 2D z uwzględnieniem typów obiektów

