

Modulacja ciągłaSygnał informacyjny: $m(t) = A_m \sin(2\pi f_m t)$

Kod programu:

```

public ChartDetails makeInformSignal (double start, double stop, double fs, double amplitude, double frequencyM){
    List<Double> scores= new ArrayList<>();
    double m=0;
    for (double t=start; t<stop;t+=(1/fs))
    {
        m=amplitude*sin(2*Math.PI*frequencyM*t);
        scores.add(m);
    }

    return new ChartDetails( title: "Inform Signal", scores, xAxisTitle: "t[s]", yAxisTitle: "m(t)");
}

```

Modulacja amplitudy: $z_A(t) = [k_A \times m(t) + 1] \times \cos(2\pi f_n t)$

Kod programu:

```

public ChartDetails makeAmplitudeModulation (double start, double stop, double fs, double kA, double frequencyN, List<Double> list)
{
    List <Double> scores= new ArrayList<>();
    int n=0;
    for (double t=start; t<stop; t+=1/fs)
    {
        double zA=0;
        zA=(kA*list.get(n)+1)*cos(2*Math.PI*frequencyN*t);
        scores.add(zA);
        n++;
    }

    return new ChartDetails( title: "Amplitude modulation", scores, xAxisTitle: "t[s]", yAxisTitle: "zA(t)" );
}

```

Modulacja fazy: $z_P(t) = \cos[2\pi f_n t + k_P \times m(t)]$

Kod programu:

```

public ChartDetails phaseModulation (double start, double stop, double fs, double kP, double fn, List<Double> list)
{
    List <Double> scores = new ArrayList<>();
    int n=0;
    double zP;
    for (double t=start; t<stop; t+=1/fs)
    {
        zP=cos(2*Math.PI*fn*t + kP*list.get(n));
        scores.add(zP);
        n++;
    }

    return new ChartDetails( title: "Phase Modulation", scores, xAxisTitle: "t[s]", yAxisTitle: "zP(t)");
}

```

Wartości początkowe:

start = 0

stop = 0.5

step = 1000

amplituda = 1

frequencyM = 10

frequencyN = 150

Zadanie 1 i 2

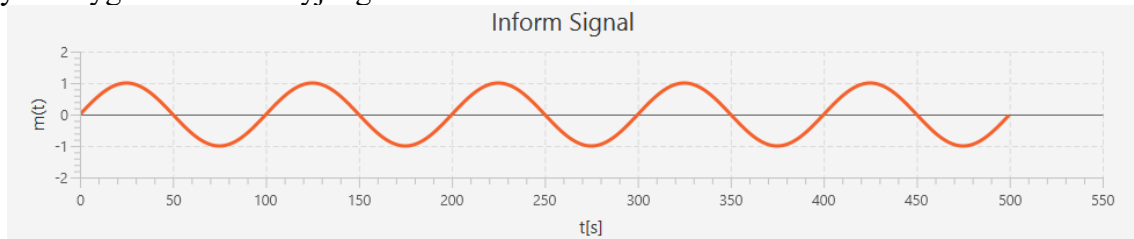
1) Wygeneruj sygnały zmodulowane $z_A(t)$ oraz $z_P(t)$ dla następujących przypadków:

- a) $1 > k_A > 0$ $k_P < 2$
- b) $12 > k_A > 2$ $\pi > k_P > 0$
- c) $k_A > BA$ $k_P > AB$

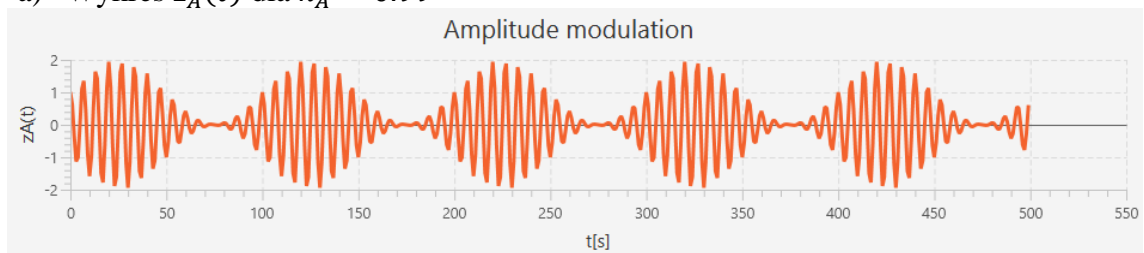
Wykonaj wykresy, w tym sygnału informacyjnego.

2) Wykonaj wykresy widm amplitudowych sygnałów zmodulowanych $z_A(t)$ oraz $z_P(t)$. Należy tak dobrać skalę (liniową lub logarytmiczną) osi poziomej i pionowej aby jak najwięcej prążków widma było widocznych na wykresie.

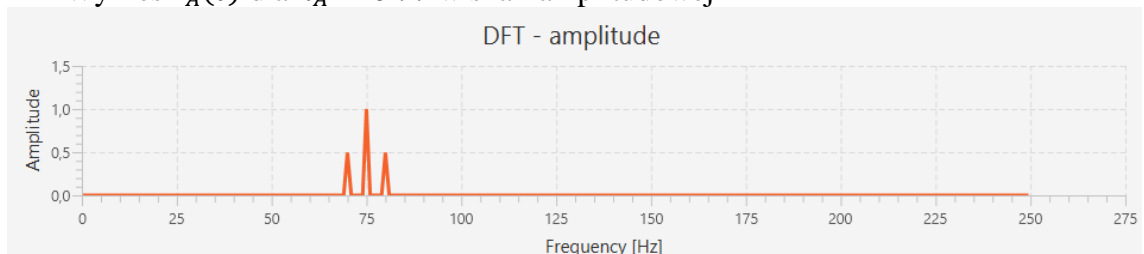
Wykres sygnału informacyjnego:



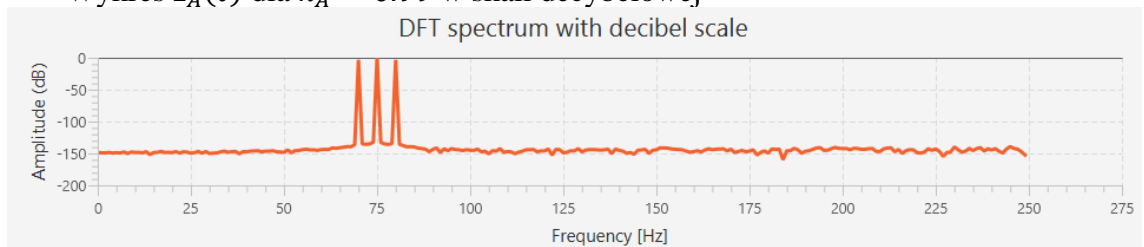
a) Wykres $z_A(t)$ dla $k_A = 0.99$

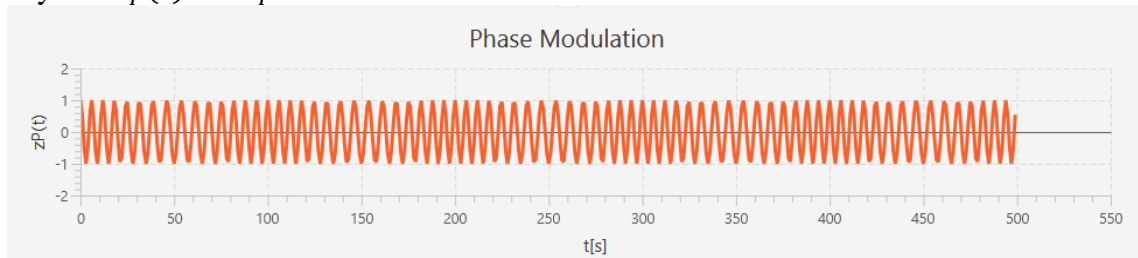
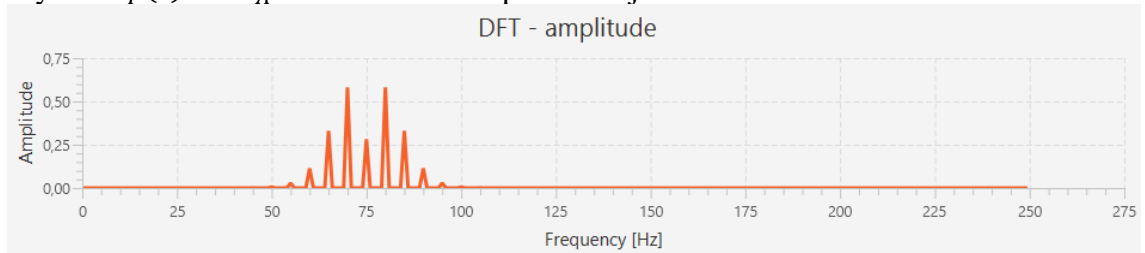
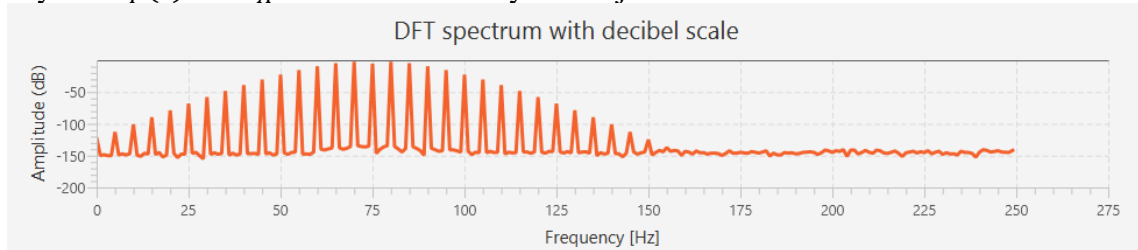
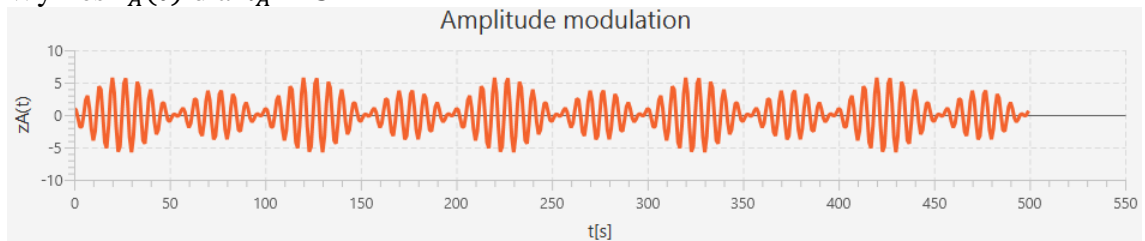
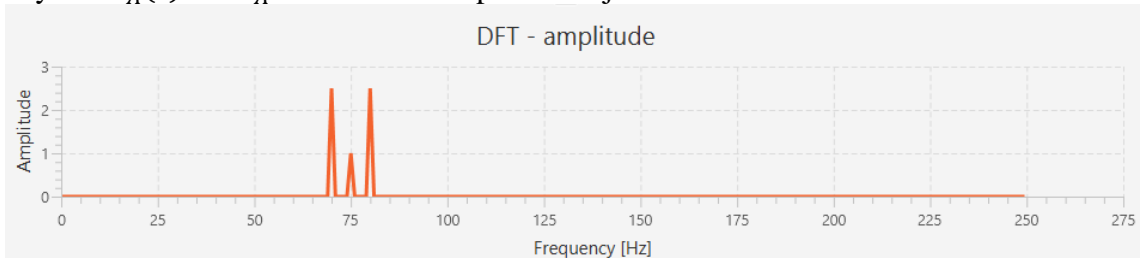


Wykres $z_A(t)$ dla $k_A = 0.99$ w skali amplitudowej

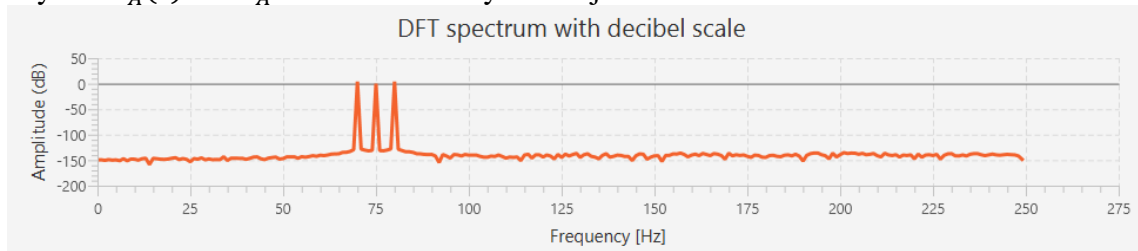


Wykres $z_A(t)$ dla $k_A = 0.99$ w skali decybelowej

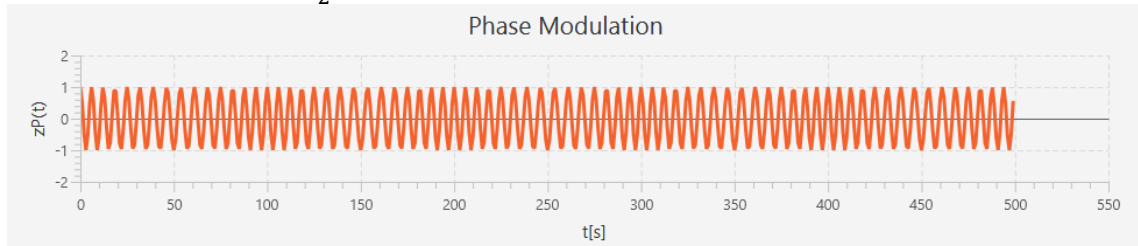


Wykres $z_P(t)$ dla $k_P = 1.9$ Wykres $z_P(t)$ dla $k_A = 1.9$ w skali amplitudowejWykres $z_P(t)$ dla $k_A = 1.9$ w skali decybelowejb) Wykres $z_A(t)$ dla $k_A = 5$ Wykres $z_A(t)$ dla $k_A = 5$ w skali amplitudowej

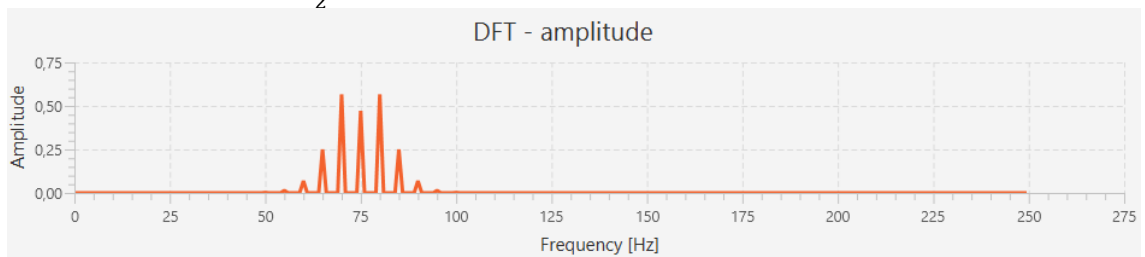
Wykres $z_A(t)$ dla $k_A = 5$ w skali decybelowej



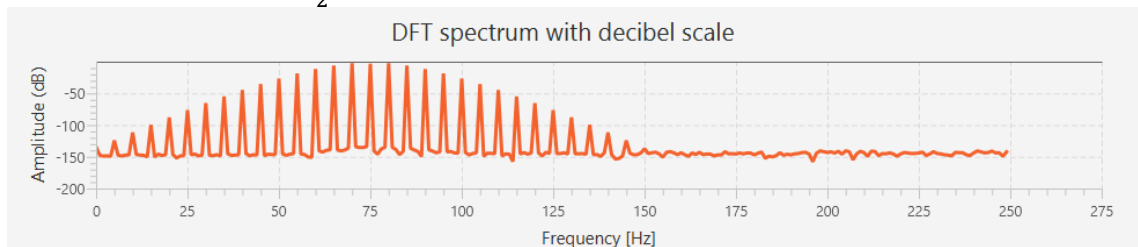
Wykres $z_P(t)$ dla $k_P = \frac{\pi}{2}$



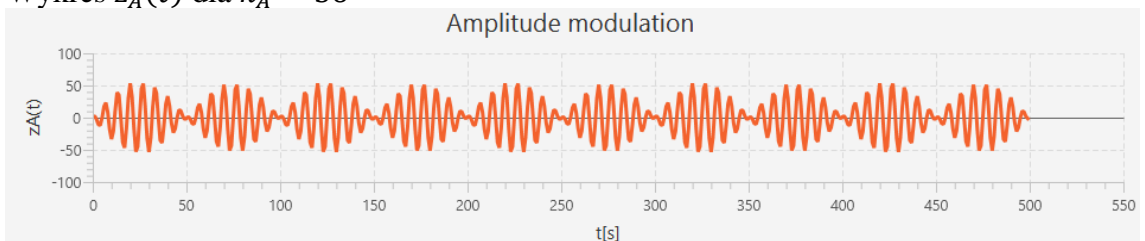
Wykres $z_P(t)$ dla $k_A = \frac{\pi}{2}$ w skali amplitudowej



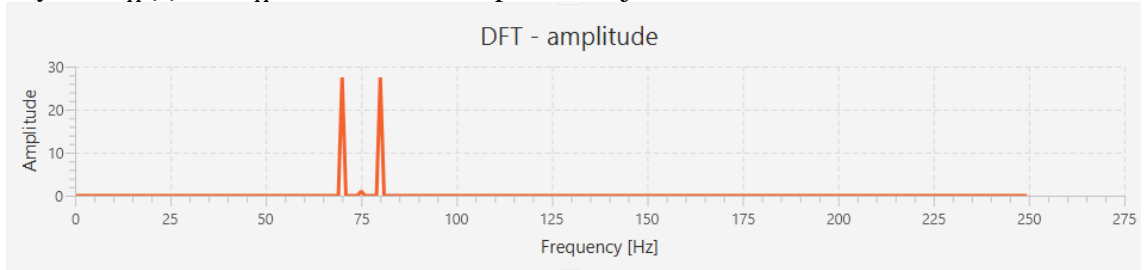
Wykres $z_P(t)$ dla $k_A = \frac{\pi}{2}$ w skali decybelowej



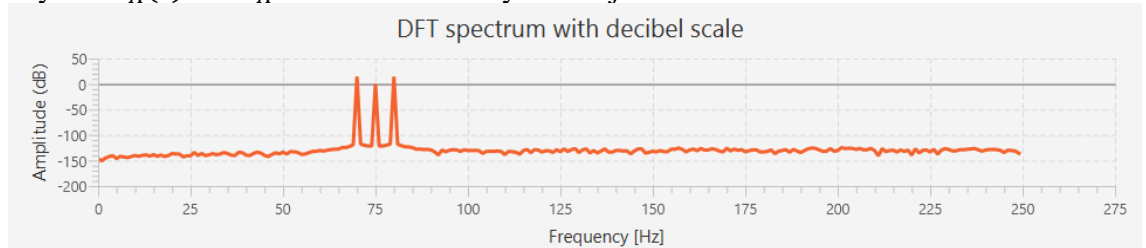
c) Wykres $z_A(t)$ dla $k_A = 56$



Wykres $z_A(t)$ dla $k_A = 56$ w skali amplitudowej

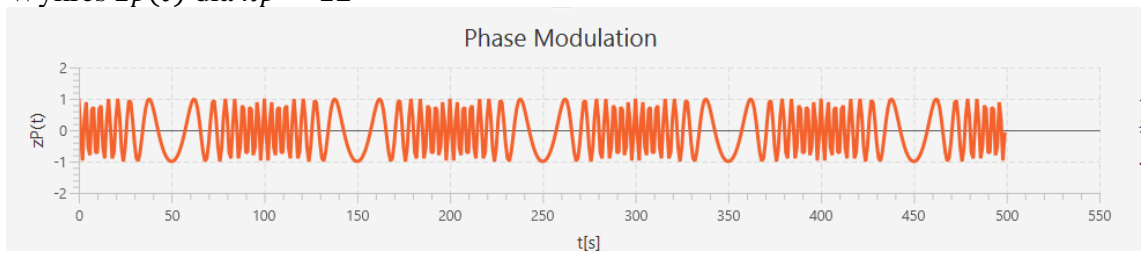


Wykres $z_A(t)$ dla $k_A = 56$ w skali decybelowej

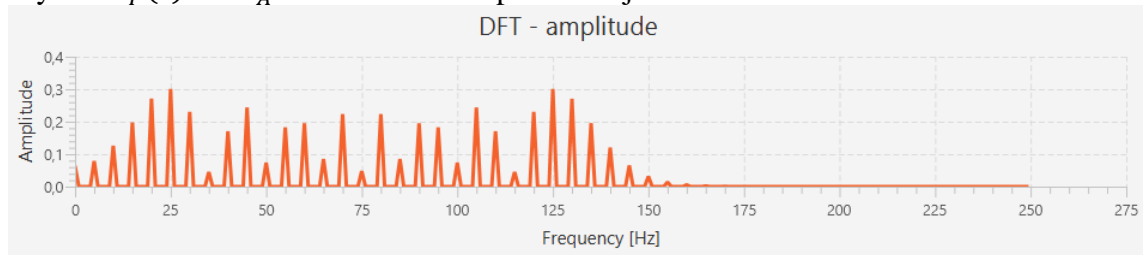


W celu lepszej widoczności wykresu $z_P(t)$ użyłam $k_P = 12$

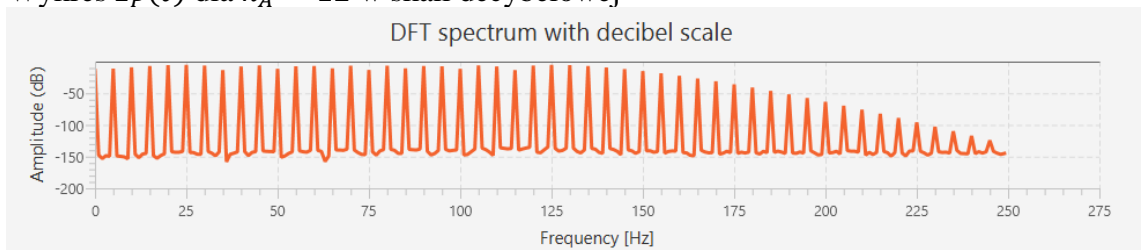
Wykres $z_P(t)$ dla $k_P = 12$



Wykres $z_P(t)$ dla $k_A = 12$ w skali amplitudowej



Wykres $z_P(t)$ dla $k_A = 12$ w skali decybelowej



Zadanie 3

Zbadaj szerokości pasma sygnałów zmodulowanych (dla poziomu -3dB) wykonując wyznaczenie granicy f_{\min} i f_{\max} oraz obliczenie $W = f_{\max} - f_{\min}$. Szerokości wyznaczonych w zadaniu pasm dla poszczególnych aproksymacji zapisz w formie komentarza w kodzie programu.

```
double signalLength(ChartDetails chartDetails, double start, double stop, double step) {  
    double max = chartDetails.getValues().stream().mapToDouble(Double::doubleValue).max().getAsDouble();  
    Map<Double, Integer> map = new LinkedHashMap<>();  
    List<Double> list = new ArrayList<>();  
    int n = 0;  
    for (Double ds : chartDetails.getValues()) {  
        double tmp = ds - max;  
        list.add(tmp);  
    }  
  
    for (Double aDouble : list) {  
        if (aDouble > -3.1) {  
            map.put(aDouble, n);  
        }  
        n++;  
    }  
    return (int) map.values().toArray()[map.size()-1] - (int) map.values().toArray()[0];  
}
```

Szerokości pasm:

Zadanie 1a)

AM – 10 Hz

PM – 20 Hz

Zadanie 1b)

AM – 10 Hz

PM – 10 Hz

Zadanie 1c)

AM – 10 Hz

PM – 120 Hz