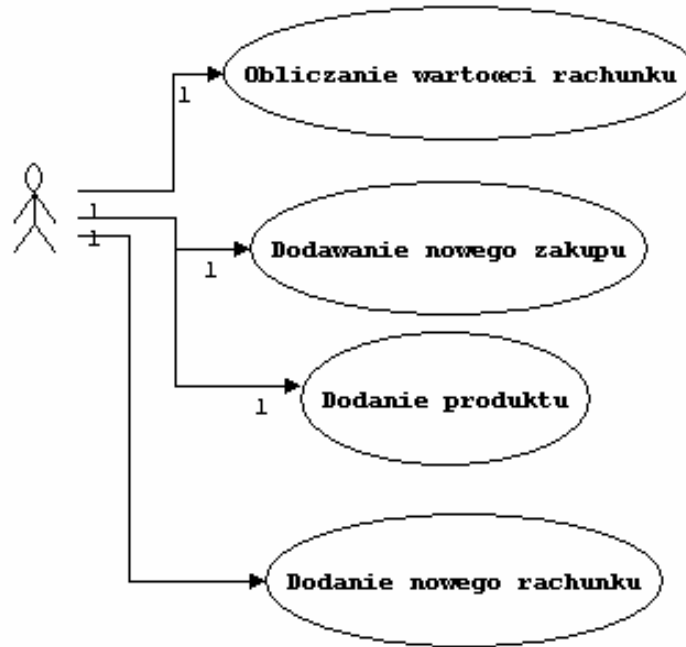


Sporządzanie rachunków

Lista wymagań

1. System zawiera katalog produktów
2. Można wprowadzić wiele rachunków
3. Pozycje rachunku muszą zawierać produkty różne w sensie typu, nazwy i ceny
4. Można zakupić trzy typy produktów różniąc się sposobem naliczania ceny: netto, z podatkiem, z promocją,
5. Każda pozycja rachunku powinna podać swoją wartość brutto oraz dane produktu oraz ilość zakupionego produktu.
6. Na rachunku powinna znajdować się wartość łączna wszystkich zakupów

Przypadki użycia systemu



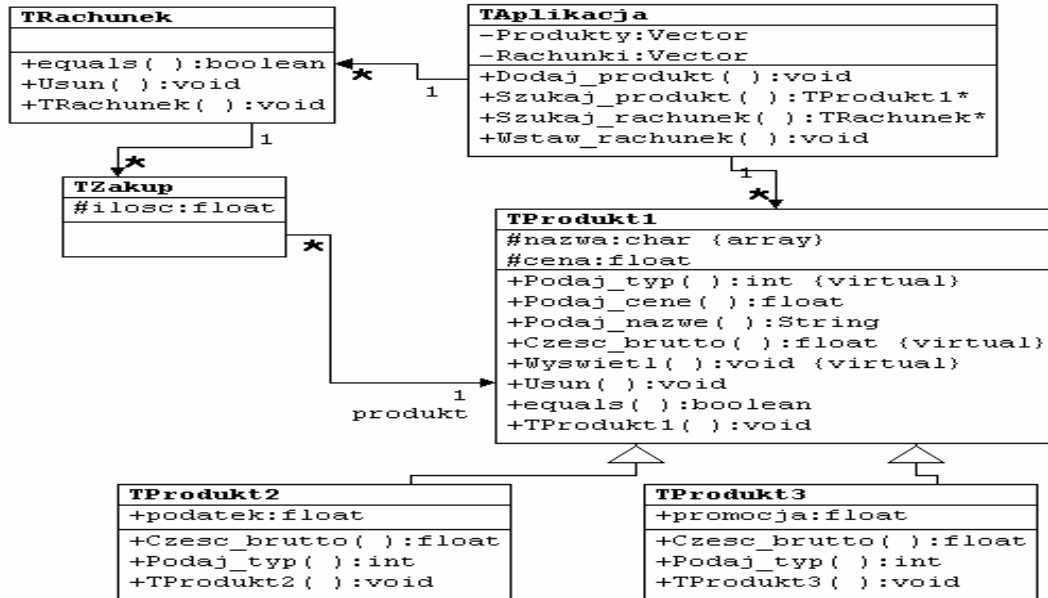
Grupa 1.

1. Wyznacz metryki: Fan-out, Fan-in, RFC na podstawie diagramu sekwencji *Dodawanie nowego zakupu*
2. Wyznacz metrykę McCabe dla następujących metod klasy *TAplikacja*: *Wstaw_zakup*, *Szukaj_rachunek*
3. Wyznacz metrykę LCOM dla klasy *TAplikacja*

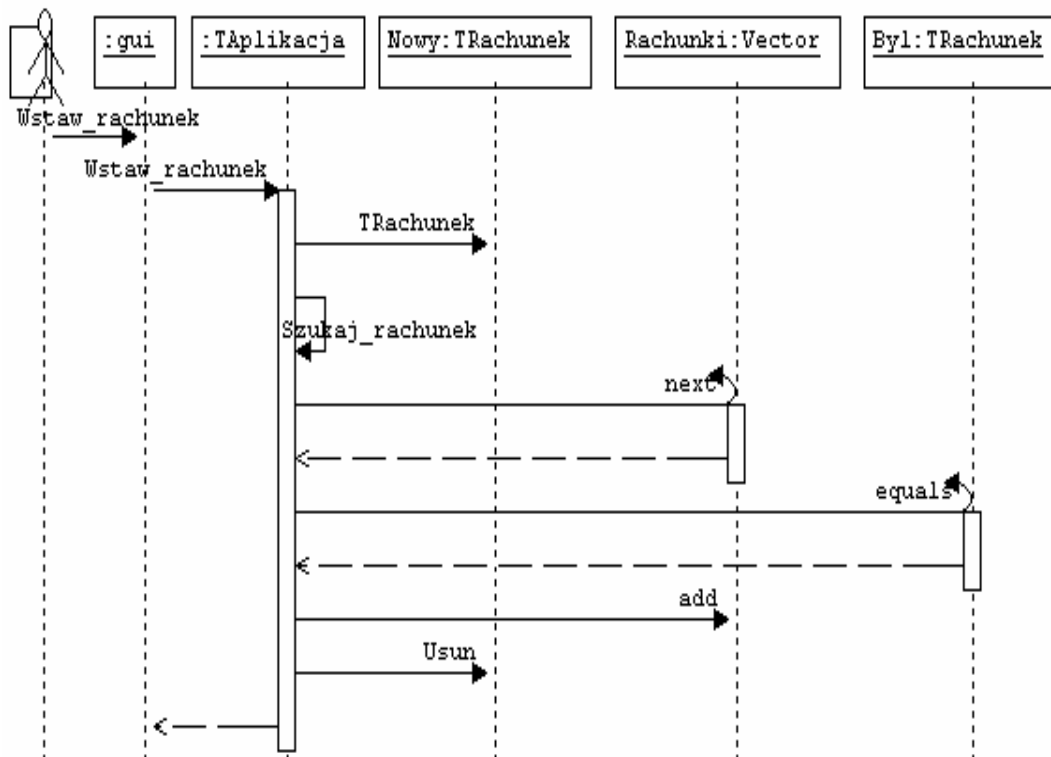
Grupa 2.

1. Wyznacz metryki: Fan-out, Fan-in, RFC na podstawie diagramu sekwencji *Dodawanie nowego rachunku*
2. Wyznacz metrykę McCabe dla następujących metod klasy *TRachunek*: *Szukaj_zakup*, *Dodaj_zakup*
3. Wyznacz metrykę LCOM dla klasy *TRachunek*

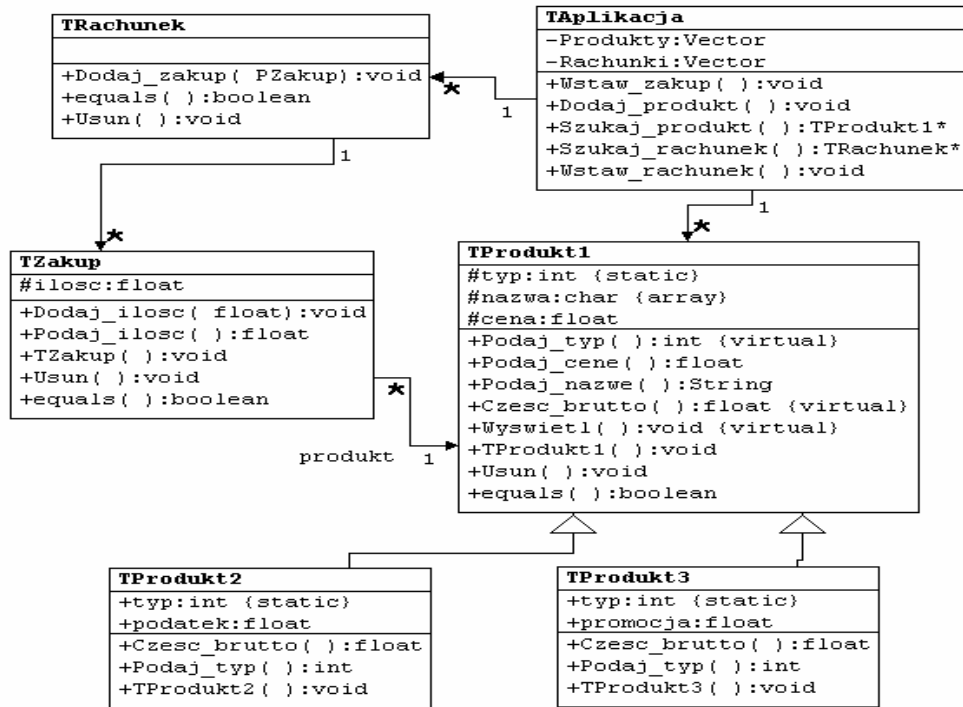
Dodawanie nowego rachunku (diagram klas) - etap projektowania2



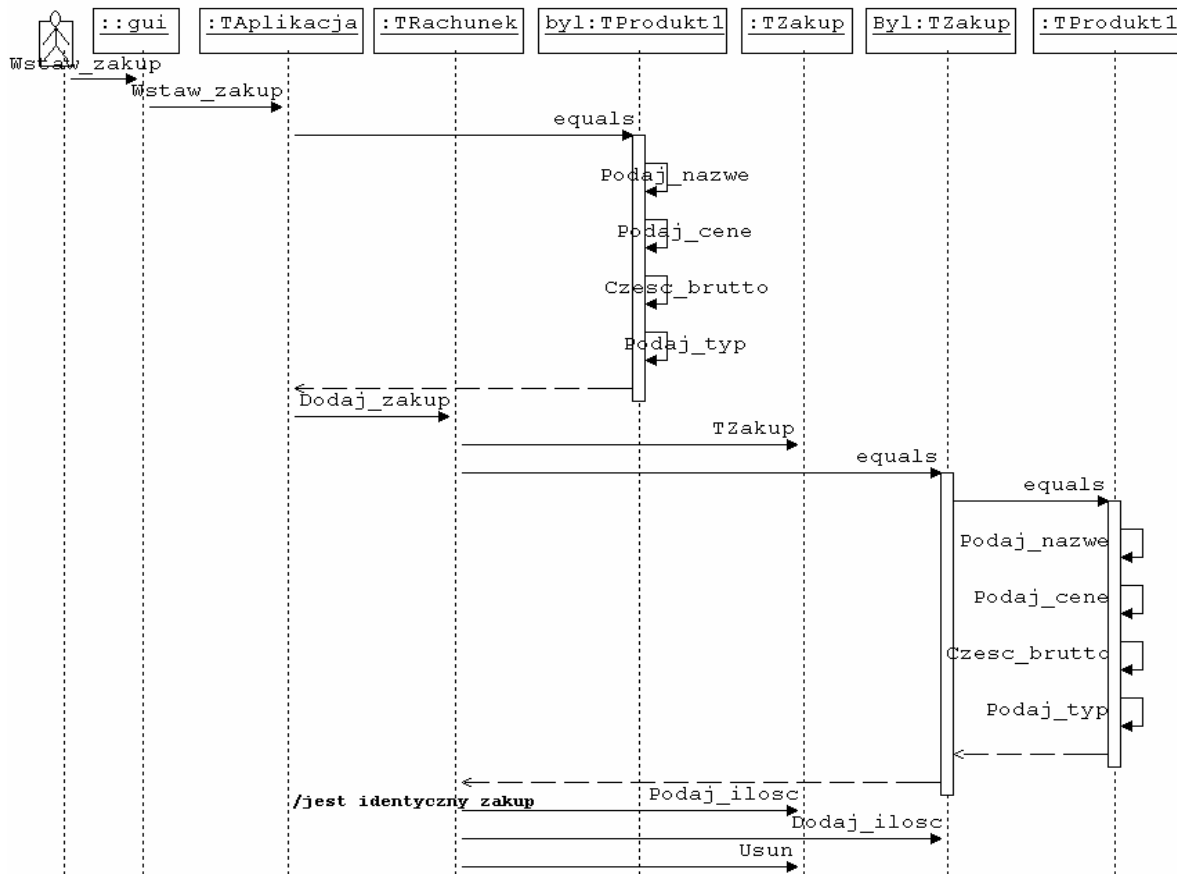
Dodawanie nowego rachunku (diagram sekwencji) - etap projektowania2



Dodawanie nowego zakupu (diagram klas) - etap analizy2



Dodawanie nowego zakupu (diagram sekwencji) - etap analizy2



Implementacja w języku JAVA

```
C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe

Produkty

nazwa : 1 cena : 1.0
nazwa : 2 cena : 2.0
nazwa : 1 cena : 1.22 podatek : 22.0
nazwa : 2 cena : 2.2 podatek : 10.0
nazwa : 1 cena : 0.78 promocja : 22.0
nazwa : 2 cena : 1.8 promocja : 10.0

Rachunek 1

ilosc : 1 Produkt : nazwa : 1 cena : 1.0
ilosc : 2 Produkt : nazwa : 1 cena : 1.22 podatek : 22.0
ilosc : 5 Produkt : nazwa : 1 cena : 0.78 promocja : 22.0
Wartosc rachunku: 7.34

Rachunek 2

ilosc : 1 Produkt : nazwa : 1 cena : 1.0
ilosc : 2 Produkt : nazwa : 1 cena : 1.22 podatek : 22.0
ilosc : 5 Produkt : nazwa : 1 cena : 0.78 promocja : 22.0
Wartosc rachunku: 7.34

Press any key to continue..._
```

```

import java.util.*;
public class TAplikacja
{ private Vector Produkty = new Vector();
  private Vector Rachunki = new Vector();

  public void Wstaw_zakup (int nr, int ile, TProdukt1 produkt1)
  { TRachunek pom;
    if ((produkt1=Szukaj_produkt(produkt1))!=null)
      if ((pom=Szukaj_rachunek(nr))!=null)
        pom.Dodaj_zakup(new TZakup(ile,produkt1));
  }
  public TRachunek Szukaj_rachunek (int nr)
  { TRachunek pom;
    Iterator it = Rachunki.iterator();
    while (it.hasNext())
      { pom = (TRachunek)it.next();
        if (pom.equals(nr))
          return pom;}
    return null; }

  public float Podaj_wartosc (int nr)
  { TRachunek pom;
    pom = Szukaj_rachunek(nr);
    if (pom!=null)
      return pom.Podaj_wartosc();
    return 0; }

  public TProdukt1 Szukaj_produkt (TProdukt1 produkt)
  { TProdukt1 pom;
    Iterator it = Produkty.iterator();
    while (it.hasNext())
      { pom = (TProdukt1)it.next();
        if (pom.equals(produkt))
          return pom;}
    return null; }

  public void Dodaj_produkt (TProdukt1 pom)
  { if (Szukaj_produkt(pom)==null)
    Produkty.add(pom);
  }
  public void Wstaw_rachunek (int nr)
  { TRachunek pom=new TRachunek(nr);
    if (Szukaj_rachunek(nr)==null)
      Rachunki.add(pom);
  }
  public void Wyswietl()
  { TProdukt1 pom;
    Iterator it = Produkty.iterator();
    while (it.hasNext())
      { pom=(TProdukt1)it.next();
        System.out.println(pom.toString()); }
  }
}

```

```

class TProdukt1
{ protected String nazwa = "";
  protected float cena = 0;
  public TProdukt1 (String anazwa, float acena )
  { super ();
    nazwa = anazwa;
    cena = acena;
  }
  public int Podaj_typ ()
  {return 0; }
  public float Podaj_cene ()
  { return cena+Czesc_brutto(); }
}

```

```

public String Podaj_nazwe ()
  { return nazwa;}
public float Czesc_brutto ()
  { return 0; }
public void Wyswietl ()
  { System.out.println(toString()); }
public String toString()
  {
  StringBuffer sb = new StringBuffer ();
  sb.append (" nazwa : " );
  sb.append ( nazwa );
  sb.append (" cena : " );
  sb.append ( Podaj_cene());
  return sb.toString ();
  }
public boolean equals ( TProdukt1 aTProdukt1 )
  { if ( aTProdukt1 == null ) return false;
    boolean bStatus = true;
    if ( Podaj_typ() != aTProdukt1.Podaj_typ() ) bStatus = false;
    else if ( !nazwa.equals( aTProdukt1.nazwa) )
      bStatus =false;
    else if (Podaj_cene()!=aTProdukt1.Podaj_cene())
      bStatus =false;
    return bStatus;
  }
}

```

```

class TProdukt2 extends TProdukt1
  { public float podatek =0 ;
  public TProdukt2 (String anazwa,float acena, float apodatek )
  { super (anazwa,acena);
    podatek = apodatek;
  }
  public String toString()
  { StringBuffer sb = new StringBuffer ();
    sb.append(super.toString());
    sb.append (" podatek : " );
    sb.append ( podatek );
    return sb.toString ();
  }
  public float Czesc_brutto ()
  { return cena*podatek/100; }
  public int Podaj_typ ()
  { return 1; }
}

```

```

class TProdukt3 extends TProdukt1
  {
  public float promocja = 0;
  public TProdukt3 (String anazwa,float acena, float apromocja )
  { super (anazwa, acena);
    promocja = apromocja;
  }
  public String toString()
  { StringBuffer sb = new StringBuffer ();
    sb.append(super.toString());
    sb.append (" promocja : " );
    sb.append ( promocja );
    return sb.toString ();
  }
  public float Czesc_brutto ()
  { return -cena*promocja/100; }
  public int Podaj_typ ()
  { return 2; }
}

```

```

class TZakup
{
    protected int ilosc = 0;
    private TProdukt1 Produkt = null;
    public TZakup ( int ailosc, TProdukt1 aProdukt )
    {
        super ();
        ilosc = ailosc;
        Produkt = aProdukt;
    }
    public String toString()
    {
        StringBuffer sb = new StringBuffer ();
        sb.append ( " ilosc : " );
        sb.append ( ilosc );
        sb.append ( " Produkt : " );
        sb.append ( Produkt.toString() );
        return sb.toString ();
    }
    public boolean equals ( TZakup aTZakup )
    {
        if ( aTZakup == null ) return false;
        boolean bStatus = true;
        if ( !Produkt.equals(aTZakup.Produkt) ) bStatus = false;
        return bStatus;
    }
    public float Podaj_wartosc ()
    { return ilosc*Produkt.Podaj_cene(); }
    public void Dodaj_ilosc ( int ailosc)
    { ilosc+=ailosc; }
    public int Podaj_ilosc ()
    { return ilosc; }
}

```

```

import java.util.*;

class TRachunek
{ int numer;
  private Vector Zakupy = new Vector();
  TRachunek (int nr)
  { numer=nr; }
  public String toString()
  { StringBuffer sb = new StringBuffer ();
    sb.append ( " Zakupy : " );
    sb.append ( numer );
    return sb.toString ();
  }
  public boolean equals ( int nr )
  { boolean bStatus = true;
    if ( numer!= nr ) bStatus = false;
    return bStatus;
  }
  public float Podaj_wartosc ()
  { float suma=0;
    TZakup pom;
    Iterator it=Zakupy.iterator();
    while (it.hasNext())
    { pom = (TZakup)it.next();
      suma += pom.Podaj_wartosc();}
    return suma;
  }
  public TZakup Szukaj_zakup(TZakup pom)
  { TZakup pom1;
    Iterator it = Zakupy.iterator();
    while (it.hasNext())
    { pom1=(TZakup)it.next();
      if (pom1.equals(pom))
        return pom1; }
    return null;
  }
  void Dodaj_zakup(TZakup pom)
  {
    TZakup pom1;
    if ((pom1=Szukaj_zakup(pom)) != null)
      pom1.Dodaj_ilosc(pom.Podaj_ilosc());
    else
      Zakupy.add(pom);
  }
  void Wyswietl()
  {
    TZakup pom;
    Iterator it = Zakupy.iterator();
    while (it.hasNext())
    { pom = (TZakup) it.next();
      System.out.println(pom.toString());
    }
  }
}

```
