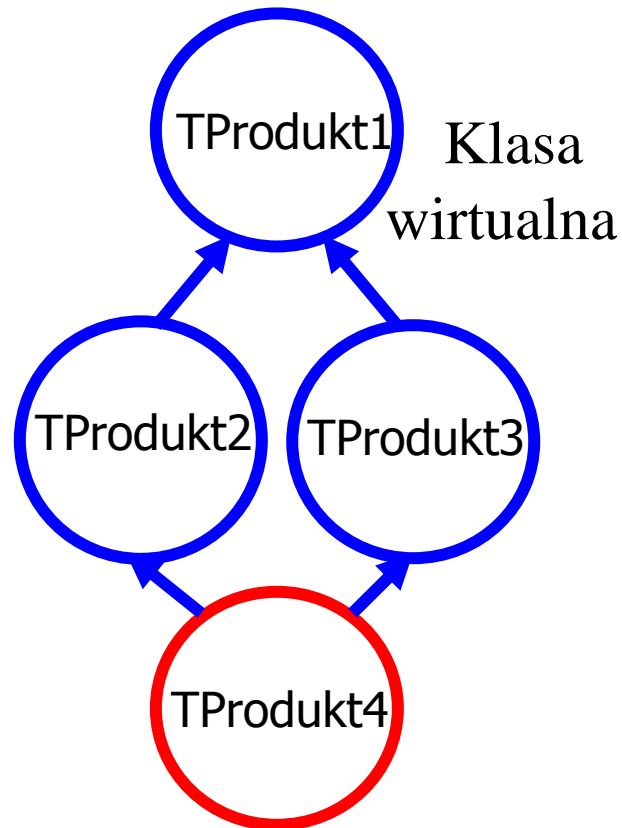
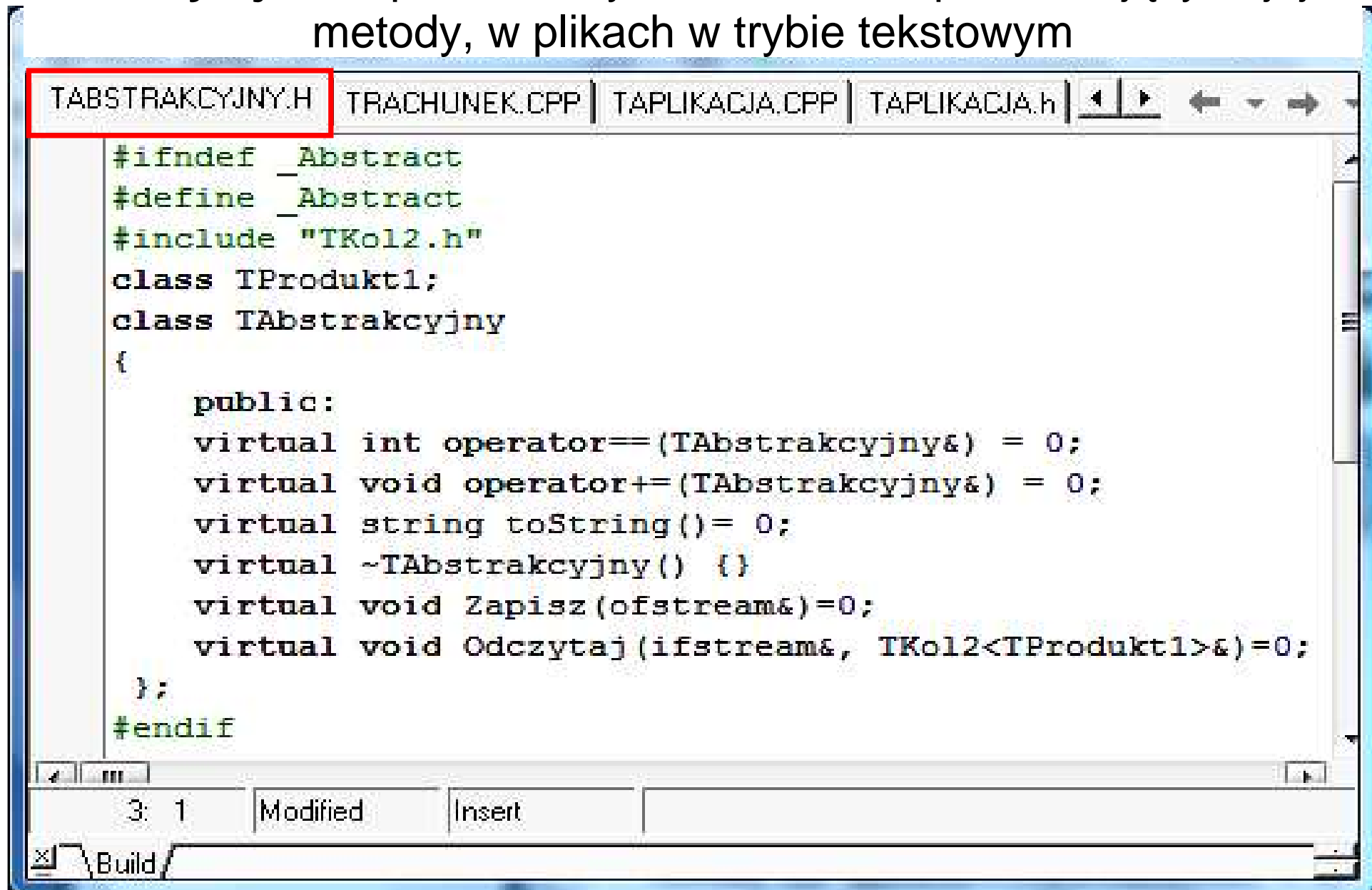


Zapis obiektów w plikach tekstowych oraz odczyt tych obiektów i zapis w pamięci - odtworzenie struktury powiązań tych obiektów

Dziedziczenie wielobazowe z powtórzeniami



Deklaracja klasy **abstrakcyjnej** dodatkowo z metodami **Zapisz** i **Odczytaj** do zapisu i odczytu obiektów, implementujących jej metody, w plikach w trybie tekstowym



```
TABSTRAKCYJNY.H | TRACHUNEK.CPP | TAPLIKACJA.CPP | TAPLIKACJA.h |
#ifndef _Abstract
#define _Abstract
#include "TKol2.h"
class TProdukt1;
class TAbstrakcyjny
{
public:
virtual int operator==(TAbstrakcyjny&) = 0;
virtual void operator+=(TAbstrakcyjny&) = 0;
virtual string toString()= 0;
virtual ~TAbstrakcyjny() {}
virtual void Zapisz(ofstream&)=0;
virtual void Odczytaj(istream&, TKol2<TProdukt1>&)=0;
};
#endif
```

3: 1 Modified Insert

Build

(1) Klasa typu pojemnik do przechowywania elementów posiadających zaimplementowane metody klasy abstrakcyjnej

```
plikrachunki1.cpp | TPRODUKT1.cpp | TPRODUKT2.CPP | TPRODUK...  
#ifndef _TKOL2  
#define _TKOL2  
#include <string.h>  
#include "TABSTRAKCYJNY.h"  
const int N=5;  
template <class T>  
class TKol2  
{protected:  
  T* kolekcja[N];  
  int ile;  
  int biezacy;  
public:  
  TKol2(int a=0)  
    {ile = a;  
     biezacy = 0; }  
  ~TKol2() { }  
  int Pusta() { return ile==0;}  
  void Zeruj() { biezacy=0;}  
  int Koniec() { return biezacy==ile; }  
  int Ile() { return ile;}  
  T*& Podaj_nast() { return kolekcja[biezacy++]; }  
};
```

(2) Klasa typu pojemnik do przechowywania elementów posiadających zaimplementowane metody klasy abstrakcyjnej

```
TZAKUP.cpp | TKOL2.h | TABSTRAKCYJNY.H | TRACHUNEK.CPP | < > <- >- >- >-
void Usun_kolekcje() {
    for (int i=0; i<ile;i++)
        delete kolekcja[i];
    ile=0;
}
int Wstaw(T* dane)
{
    for (int i=0;i<ile;i++)
        if (*dane == *kolekcja[i])
            { *(kolekcja[i])+=*dane;
              delete dane;    return 1;}
    if(ile==N)
        { delete dane;    return 0; }
    kolekcja[ile++]=dane;
    return 2;
}
T* Podaj(T* dane )
{ T* pom=NULL;
  for (int i=0;i<ile;i++)
      if (*dane == *kolekcja[i])
          { pom= kolekcja[i];
            break; }
  return pom;}

```

(3) Klasa typu pojemnik do przechowywania elementów posiadających zaimplementowane metody klasy abstrakcyjnej

```
TZAKUP.cpp TKOL2.h TABSTRAKCYJNY.H TRACHUNEK.CPP TAPL
string toString()
{ string s;
  for (int i=0; i<ile;i++)
    s+=kolekcja[i]->toString()+"\n";
  return s; }

friend ostream& operator<<(ostream& wy, TKol2<T>& kol)
{ return wy<<kol.toString()<<endl; }

};
#endif

51: 1 Modified Insert
Build
```

(1) Deklaracja klasy należącej do rodziny klas dziedziczących wielobazowo z klasą wirtualną **TProdukt1**– klasa **TProdukt1** dodatkowo z metodami **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym

```
TPRODUKT1.cpp | TPRODUKT1.h | TPRODUKT2.CPP | TPRODUKT3.CPP | TPRODU...  
#ifndef _Produkt1  
#define _Produkt1  
#include <iostream.h>  
#include <fstream.h>  
#include <string.h>  
#include <stdlib.h>  
#include "TABSTRAKCYJNY.h"  
class TProdukt1: public TAbstrakcyjny  
{protected:  
    string nazwa;  
    float cena;  
public:  
    TProdukt1(string nazwa_="bez nazwy",float cena_=0);  
    TProdukt1(TProdukt1&);  
    ~TProdukt1();  
    string Podaj_nazwe() {return nazwa;}//dodano po dodaniu GUI  
    virtual float Podaj_cene();  
    virtual float Podaj_podatek() { return -1; }  
    virtual float Podaj_promocje() { return -1; }  
    void operator+=(TAbstrakcyjny&){}  
    int operator==(TAbstrakcyjny&);  
    string toString();  
    friend ostream& operator<<(ostream&, TProdukt1&);  
    void zapisz(ofstream&);  
    void Zapisz(ofstream&);  
    void odczytaj(ifstream&);  
    void Odczytaj(ifstream&, TKol2 <TProdukt1>&);  
};  
#endif
```

Definicje metod **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym (przy odczycie obiekty są odtwarzane i zapisywane w pamięci) - **klasa TProdukt1**

TPRODUKT1.cpp

TPRODUKT1.h

TPRODUKT2.CPP

TPRODUKT3.CPP

TPRODU

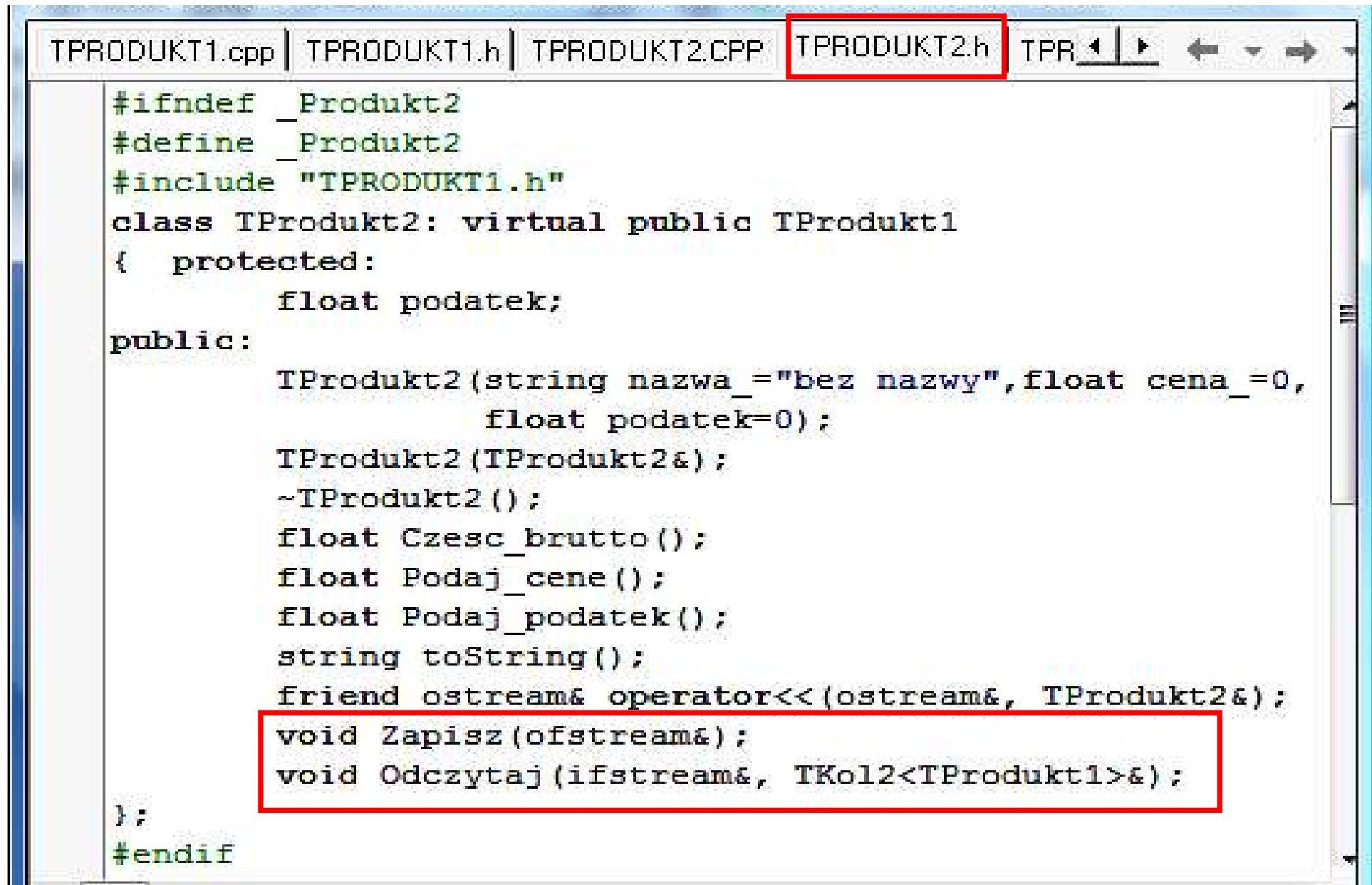
```
void TProdukt1::zapisz(ofstream& pwy)
{
    pwy<<nazwa<<endl;
    pwy<<cena<<endl;
}

void TProdukt1::Zapisz(ofstream& pwy)
{ pwy<<"<TProdukt1>"<<endl;
  zapisz(pwy);
  pwy<<"</TProdukt1>"<<endl;
}

void TProdukt1::odczytaj(ifstream& pwe)
{
    pwe>>nazwa;
    pwe>>cena;
}

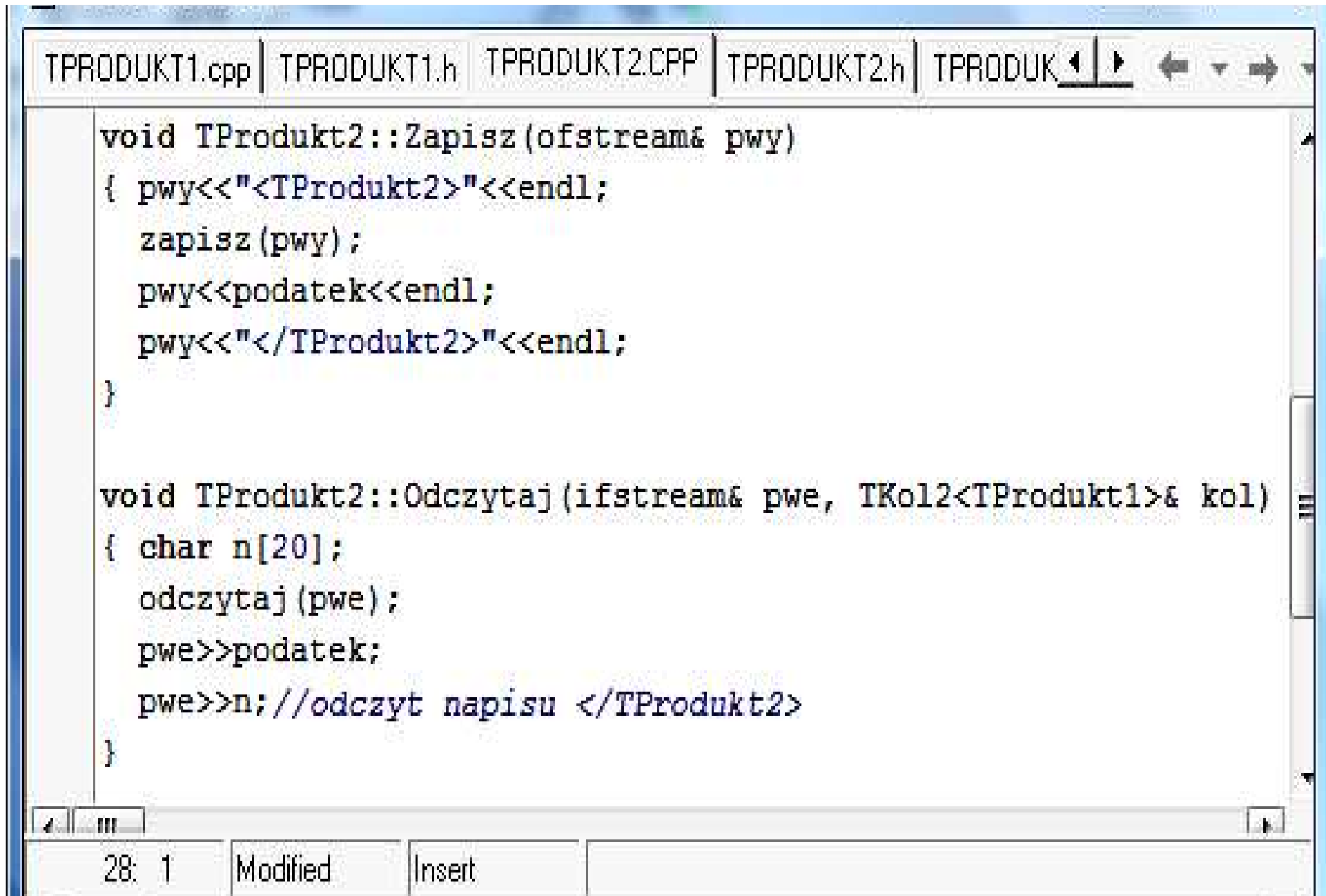
void TProdukt1::Odczytaj(ifstream& pwe, TKol2<TProdukt1>& kol)
{ char n[20];
  odczytaj(pwe);
  pwe>>n;//odczyt napisu </TProdukt1>
}
```


(2) Deklaracja klasy należącej do rodziny klas dziedziczących wielobazowo z klasą wirtualną **TProdukt1** – klasa **TProdukt2** dodatkowo z metodami **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym



```
TPRODUKT1.cpp | TPRODUKT1.h | TPRODUKT2.CPP | TPRODUKT2.h | TPR...  
#ifndef _Produkt2  
#define _Produkt2  
#include "TPRODUKT1.h"  
class TProdukt2: virtual public TProdukt1  
{ protected:  
    float podatek;  
public:  
    TProdukt2 (string nazwa_="bez nazwy", float cena_=0,  
                float podatek=0);  
    TProdukt2 (TProdukt2&);  
    ~TProdukt2 ();  
    float Czesc_brutto ();  
    float Podaj_cene ();  
    float Podaj_podatek ();  
    string toString ();  
    friend ostream& operator<< (ostream&, TProdukt2&);  
    void Zapisz (ofstream&);  
    void Odczytaj (ifstream&, TKol2<TProdukt1>&);  
};  
#endif
```

Definicje metod **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym (przy odczycie obiekty są odtwarzane i zapisywane w pamięci) - **klasa TProdukt2**



The image shows a screenshot of a code editor window with a tabbed interface. The active tab is 'TPRODUKT2.CPP'. The code defines two methods for the TProdukt2 class: Zapisz and Odczytaj. The Zapisz method writes the object's data to an output stream, and the Odczytaj method reads data from an input stream and reconstructs the object.

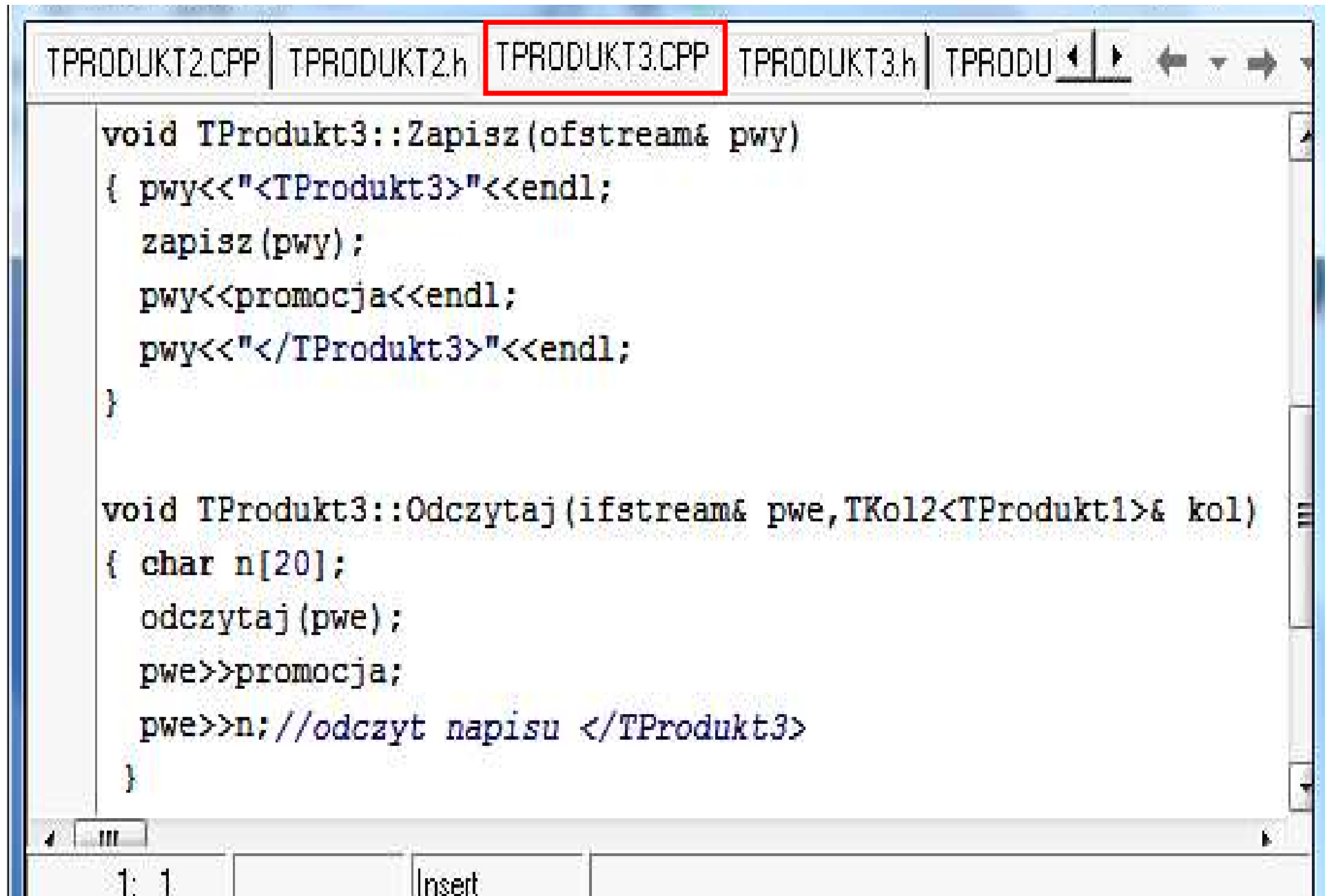
```
TPRODUKT1.cpp | TPRODUKT1.h | TPRODUKT2.CPP | TPRODUKT2.h | TPRODUK...  
  
void TProdukt2::Zapisz(ofstream& pwy)  
{ pwy<<"<TProdukt2>"<<endl;  
  zapisz(pwy);  
  pwy<<podatek<<endl;  
  pwy<<"</TProdukt2>"<<endl;  
}  
  
void TProdukt2::Odczytaj(ifstream& pwe, TKol2<TProdukt1>& kol)  
{ char n[20];  
  odczytaj(pwe);  
  pwe>>podatek;  
  pwe>>n;//odczyt napisu </TProdukt2>  
}
```

28: 1 Modified Insert

(3) Deklaracja klasy należącej do rodziny klas dziedziczących wielobazowo z klasą wirtualną **TProdukt1**– klasa **TProdukt3** dodatkowo z metodami **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym

```
TPRODUKT2.CPP | TPRODUKT2.h | TPRODUKT3.CPP | TPRODUKT3.h | TPRODUK...  
#ifndef _Produkt3  
#define _Produkt3  
#include "TPRODUKT1.h"  
class TProdukt3: virtual public TProdukt1  
{ protected:  
    float promocja;  
public:  
    TProdukt3(string nazwa_="bez nazwy",float cena_=0,  
              float promocja=0);  
    TProdukt3(TProdukt3&);  
    ~TProdukt3();  
    float Czesc_brutto();  
    float Podaj_cene();  
    float Podaj_promocje();  
    string toString();  
    friend ostream& operator<<(ostream& wy, TProdukt3& p);  
    void Zapisz(ofstream&);  
    void Odczytaj(ifstream&, TKol2<TProdukt1>&);  
};  
#endif
```

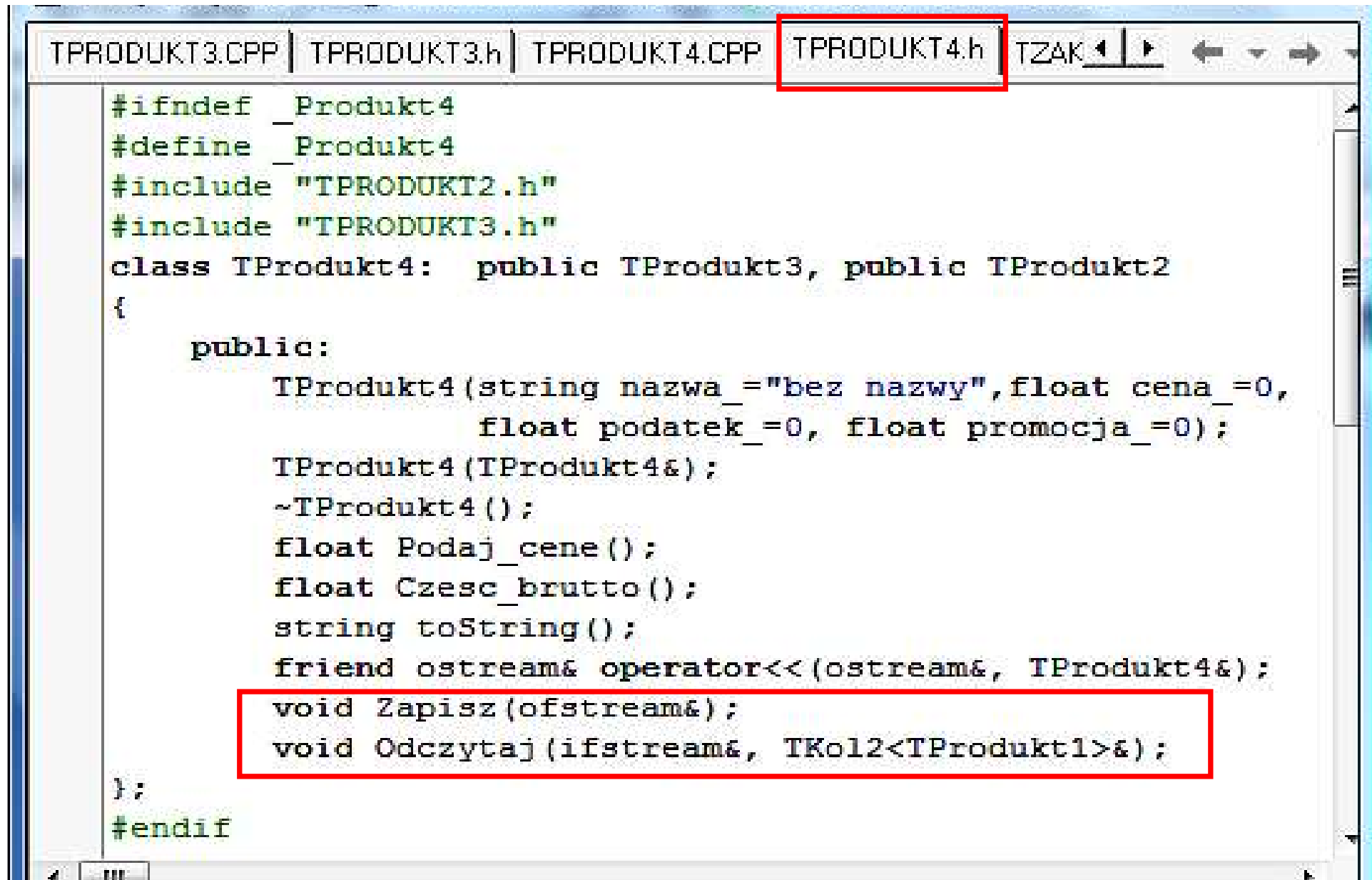
Definicje metod **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym (przy odczycie obiekty są odtwarzane i zapisywane w pamięci) - **klasa TProdukt3**



```
TPRODUKT2.CPP | TPRODUKT2.h | TPRODUKT3.CPP | TPRODUKT3.h | TPRODU...  
  
void TProdukt3::Zapisz(ofstream& pwy)  
{ pwy<<"<TProdukt3>"<<endl;  
  zapisz(pwy);  
  pwy<<"promocja"<<endl;  
  pwy<<"</TProdukt3>"<<endl;  
}  
  
void TProdukt3::Odczytaj(istream& pwe,TKol2<TProdukt1>& kol)  
{ char n[20];  
  odczytaj(pwe);  
  pwe>>"promocja";  
  pwe>>n;//odczyt napisu </TProdukt3>  
}
```

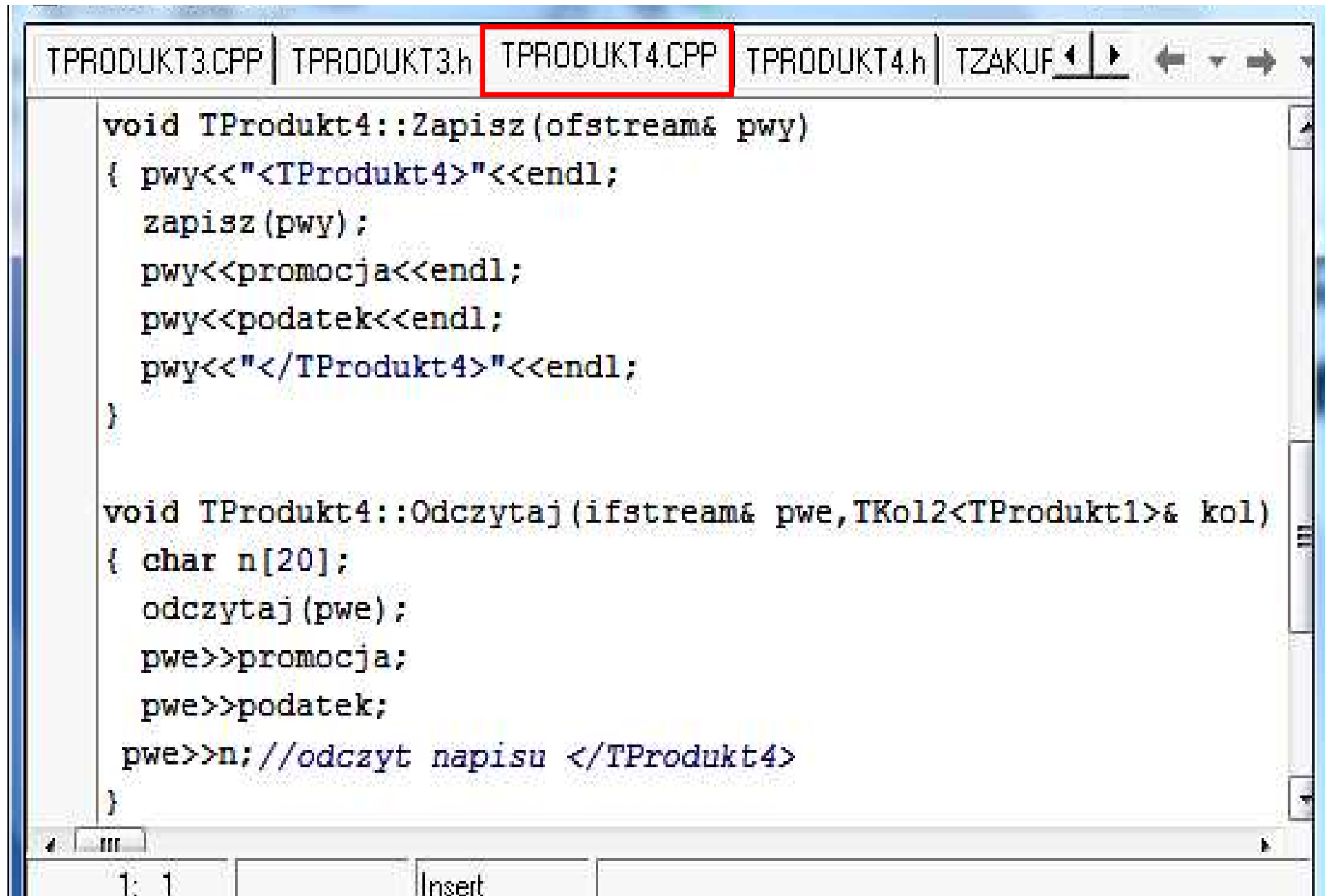
1: 1 | Insert

(4) Deklaracja klasy należącej do rodziny klas dziedziczących wielobazowo z klasą wirtualną **TProdukt1**– klasa **TProdukt4** dodatkowo z metodami **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym



```
TPRODUKT3.CPP | TPRODUKT3.h | TPRODUKT4.CPP | TPRODUKT4.h | TZAK ◀ ▶ ◀ ▶ ◀ ▶
#ifndef _Produkt4
#define _Produkt4
#include "TPRODUKT2.h"
#include "TPRODUKT3.h"
class TProdukt4: public TProdukt3, public TProdukt2
{
public:
    TProdukt4(string nazwa_="bez nazwy",float cena_=0,
              float podatek_=0, float promocja_=0);
    TProdukt4(TProdukt4&);
    ~TProdukt4();
    float Podaj_cene();
    float Czesc_brutto();
    string toString();
    friend ostream& operator<<(ostream&, TProdukt4&);
    void Zapisz(ofstream&);
    void Odczytaj(ifstream&, TKol2<TProdukt1>&);
};
#endif
```

Definicje metod **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym (przy odczycie obiekty są odtwarzane i zapisywane w pamięci) - **klasa TProdukt4**



The image shows a screenshot of a code editor window with several tabs at the top: TPRODUKT3.CPP, TPRODUKT3.h, TPRODUKT4.CPP (highlighted with a red box), TPRODUKT4.h, and TZAKUF. The main area contains C++ code for the TProdukt4 class. The code defines two methods: Zapisz and Odczytaj. The Zapisz method writes the object's data to an output stream, and the Odczytaj method reads the data from an input stream and reconstructs the object.

```
void TProdukt4::Zapisz(ofstream& pwy)
{ pwy<<"<TProdukt4>"<<endl;
  zapisz(pwy);
  pwy<<promocja<<endl;
  pwy<<podatek<<endl;
  pwy<<"</TProdukt4>"<<endl;
}

void TProdukt4::Odczytaj(istream& pwe,TKol2<TProdukt1>& kol)
{ char n[20];
  odczytaj(pwe);
  pwe>>promocja;
  pwe>>podatek;
  pwe>>n;//odczyt napisu </TProdukt4>
}
```

At the bottom of the editor, the status bar shows "1: 1" and "Insert" mode.

(5) Deklaracja klasy **TZakup** dodatkowo z metodami **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym

```
TPRODUKT4.CPP | TPRODUKT4.h | TZAKUP.cpp | TZAKUP.h | TK <|> <- -> >
#ifndef _Zakup
#define _Zakup
#include "TPRODUKT4.h"
class TZakup: public TAbstrakcyjny
{
    protected:
        TProdukt1* produkt;
        float ilosc;
    public:
        TZakup(TProdukt1*produkt_=NULL,float ilosc_=0);
        TZakup(TZakup&);
        ~TZakup();
        float Podaj_wartosc();
        float Podaj_ilosc();
        TProdukt1* Podaj_produkt() { return produkt;}
        void operator+=(TAbstrakcyjny&);
        int operator==(TAbstrakcyjny&);
        string toString();
        friend ostream& operator<<(ostream&, TZakup&);
        void Zapisz(ofstream&);
        void Odczytaj(ifstream&, TKol2<TProdukt1>&);
};
#endif
```

Definicje metod **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym (przy odczycie obiekty są odtwarzane i zapisywane w pamięci) - **klasa TZakup**

```
TPRODUKT4.CPP | TPRODUKT4.h | TZAKUP.cpp | TZAKUP.h | TKOL2.h | TAE | < | > | < | > | < | > | < | >
void TZakup::Zapisz(ofstream& pwy)
{
    pwy<<"<TZakup>"<<endl;
    pwy<<ilosc<<endl;
    produkt->Zapisz(pwy);
    pwy<<"</TZakup>"<<endl;
}

void TZakup::Odczytaj(ifstream& pwe, TKol2<TProdukt1>& kol)
{
    TProdukt1* pom;
    char n[20];
    pwe>>n;//odczyt napisu <TZakup>
    pwe>>ilosc;
    pwe>>n; //odczyt napisu <TProduktx>
    if (!strcmp(n, "<TProdukt1>"))        pom = new TProdukt1;
    else
        if (!strcmp(n, "<TProdukt2>"))    pom = new TProdukt2;
    else
        if (!strcmp(n, "<TProdukt3>"))    pom = new TProdukt3;
    else
        if (!strcmp(n, "<TProdukt4>"))    pom = new TProdukt4;
    if (pom!=NULL) pom->Odczytaj(pwe, kol);
    produkt = kol.Podaj(pom);
    delete pom;
    pwe>>n;//odczyt napisu </TZakup>
}
}
```


(6) Deklaracje klasy TRachunek dodatkowo z metodami Zapisz i Odczytaj do zapisu i odczytu obiektów w plikach w trybie tekstowym

```
TABSTRAKCYJNY.H | TRACHUNEK.CPP | TRACHUNEK.h | TAPLIKA < > < > < > < >
#ifdef _Rachunek
#define _Rachunek
#include "TZakup.h"
#include "TKOL2.h"
class TRachunek:public TAbstrakcyjny
{protected:
    int numer;
    TKol2<TZakup> zakupy;
    void kopia(TRachunek& r);
public:
    TRachunek(int=0);
    TRachunek(TRachunek& r);
    ~TRachunek();
    TKol2<TZakup>& Podaj_zakupy() { return zakupy;}
    int Podaj_numer()           { return numer; }
    void operator=(TRachunek&);
    float Podaj_wartosc();
    int Dodaj_zakup(TZakup*);
    void operator+=(TAbstrakcyjny&) {}
    int operator==(TAbstrakcyjny&);
    string toString();
    friend ostream& operator<<(ostream&, TRachunek&);
    void Zapisz(ofstream&);
    void Odczytaj(ifstream&, TKol2<TProdukt1>&);
};
#endif
```

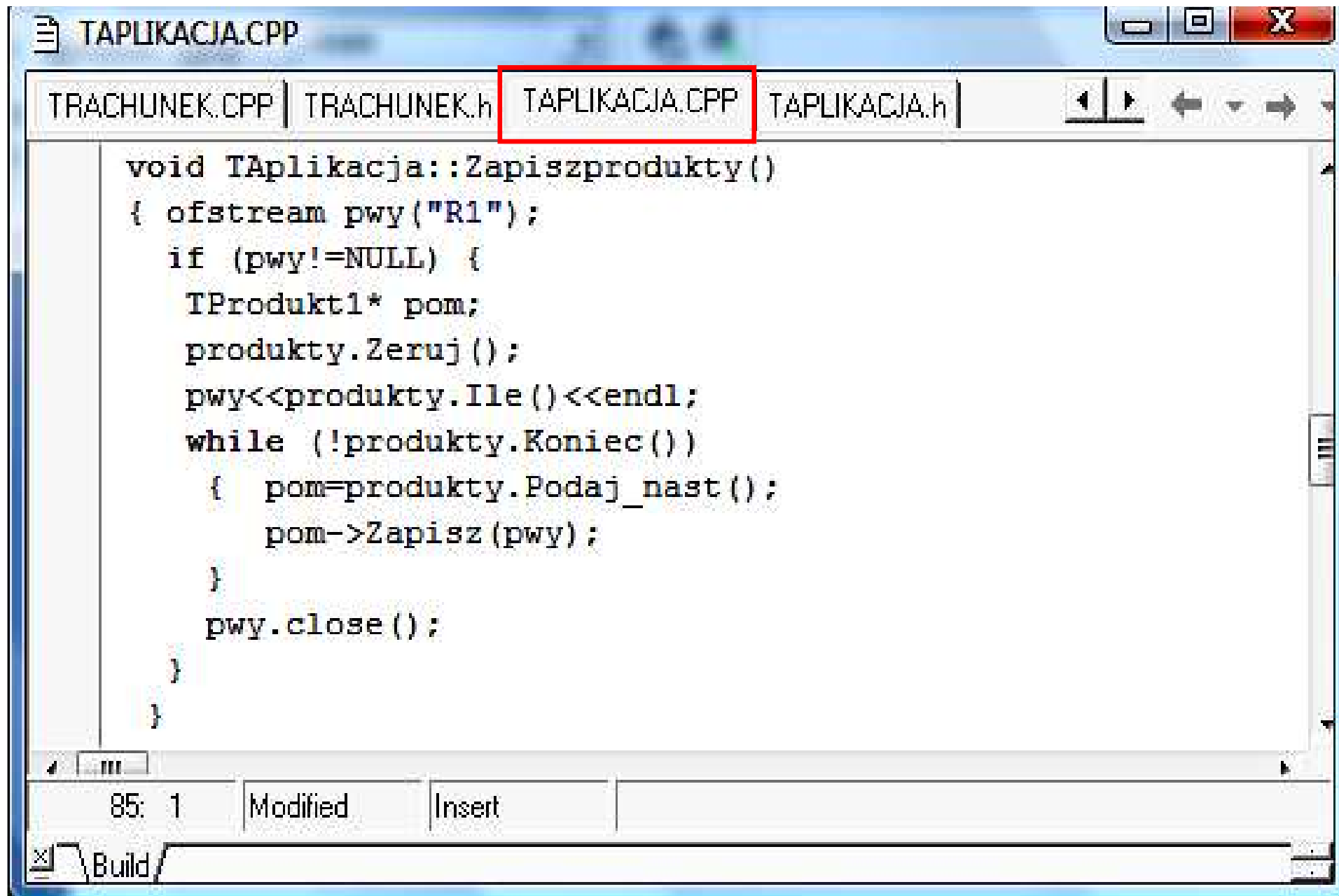
Definicje metod **Zapisz** i **Odczytaj** do zapisu i odczytu w plikach w trybie tekstowym (przy odczycie obiekty są odtwarzane i zapisywane w pamięci) - **klasa TRachunek**

```
TABSTRAKCYJNY.H | TRACHUNEK.CPP | TRACHUNEK.H | TAPLIKACJA.CPP | TAPLIKA...  
void TRachunek::Zapisz(ofstream& pwy)  
{  
    TZakup* z;  
    zakupy.Zeruj();  
    pwy<<"<TRachunek>"<<endl;  
    pwy<<numer<<endl;  
    pwy<<zakupy.Ile()<<endl;  
    while(!zakupy.Koniec())  
        { z=(TZakup*)zakupy.Podaj_nast();  
          z->Zapisz(pwy);  
        }  
    pwy<<"</TRachunek>"<<endl;  
}  
void TRachunek::Odczytaj(ifstream& pwe, TKol2<TProdukt1>& kol)  
{ char n[20];  
  TZakup* z;  
  zakupy.Usun_kolekcje();  
  int ile;  
  pwe>>n;//odczyt napisu <TRachunek>  
  pwe>>numer;  
  pwe>>ile;  
  while(ile--)  
      { z=new TZakup;  
        z->Odczytaj(pwe,kol);  
        zakupy.Wstaw(z);  
      }  
  pwe>>n;//odczyt napisu </TRachunek>  
}
```

(7) Deklaracja klasy **TAplikacja** dodatkowo z metodami **Zapiszprodukty**, **Zapiszrachunki** i **Odczytajprodukty** oraz **Odczytajrachunki** do zapisu i odczytu pojemnika z produktami i pojemnika z rachunkami w plikach w trybie tekstowym

```
TRACHUNEK.CPP | TRACHUNEK.h | TAPLIKACJA.CPP | TAPLIKACJA.h
#ifndef _Aplikacja
#define _Aplikacja
#include "TRACHUNEK.h"
class TAplikacja
{ TKol2<TProdukt1> produkty;
  TKol2<TRachunek> rachunki;
  friend class TZakup;
  TProdukt1* Wykonaj_produkt(string* atrybuty);
public:
  ~TAplikacja()
  { produkty.Usun_kolekcje();
    rachunki.Usun_kolekcje();
  }
  int Wstaw_zakup(string* produkt,int ilosc, int numer);
  int Wstaw_produkt(string* wstawianyprodukt);
  int Wstaw_rachunek(int nr);
  TKol2<TProdukt1>& Podaj_produkty();
  TKol2<TRachunek>& Podaj_rachunki();
  TRachunek* Podaj_rachunek(int nr);
  void Zapiszprodukty();
  void Odczytajprodukty();
  void Zapiszrachunki();
  void Odczytajrachunki();
};
#endif
```

Definicja metody **Zapiszprodukty** do zapisu kolekcji różnych typów produktów w pliku o nazwie „R1” w trybie tekstowym - **klasa TApplikacja**

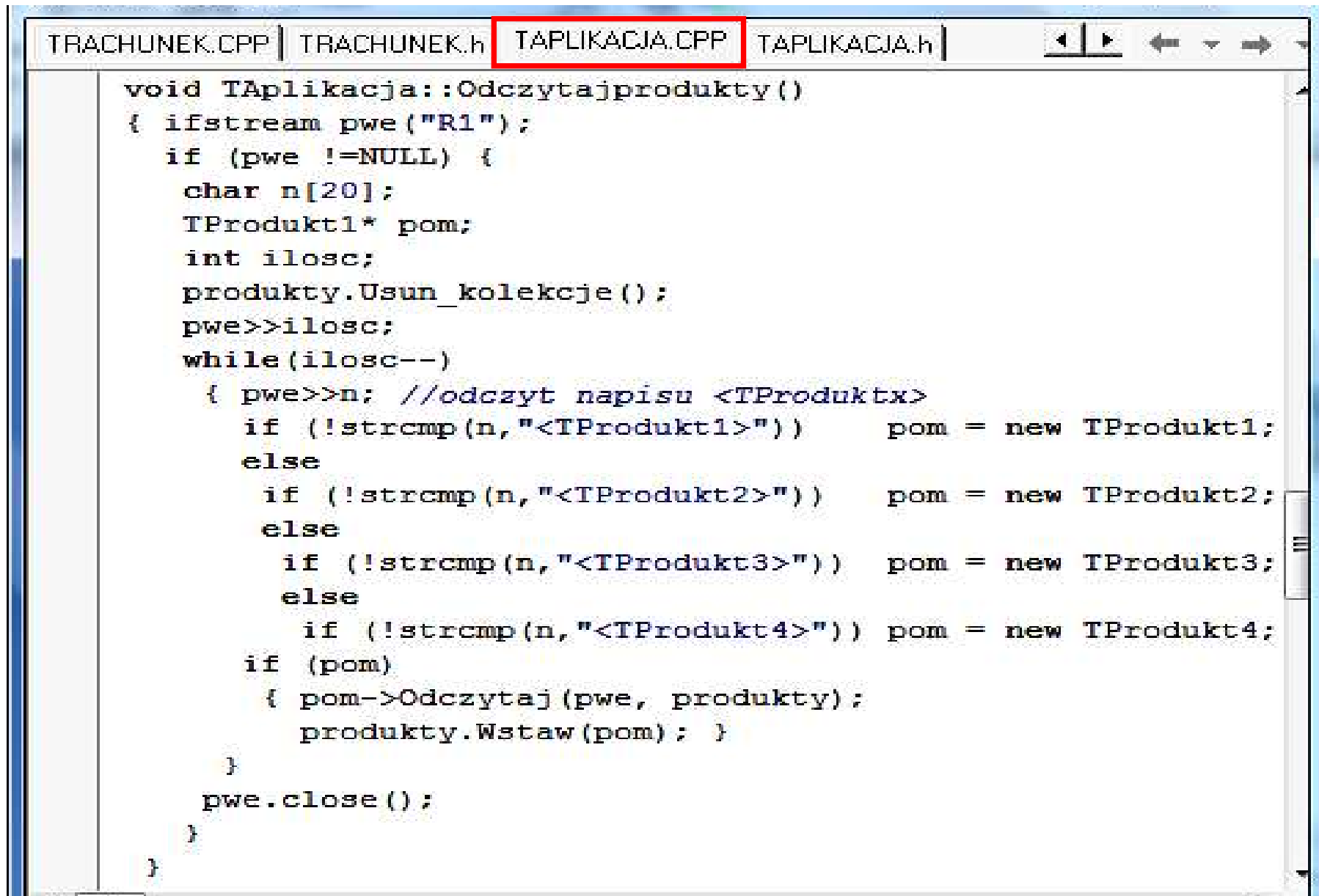


```
void TApplikacja::Zapiszprodukty()
{ ofstream pwy("R1");
  if (pwy!=NULL) {
    TProdukt1* pom;
    produkty.Zeruj();
    pwy<<produkty.Ile()<<endl;
    while (!produkty.Koniec())
      { pom=produkty.Podaj_nast();
        pom->Zapisz(pwy);
      }
    pwy.close();
  }
}
```

85: 1 Modified Insert

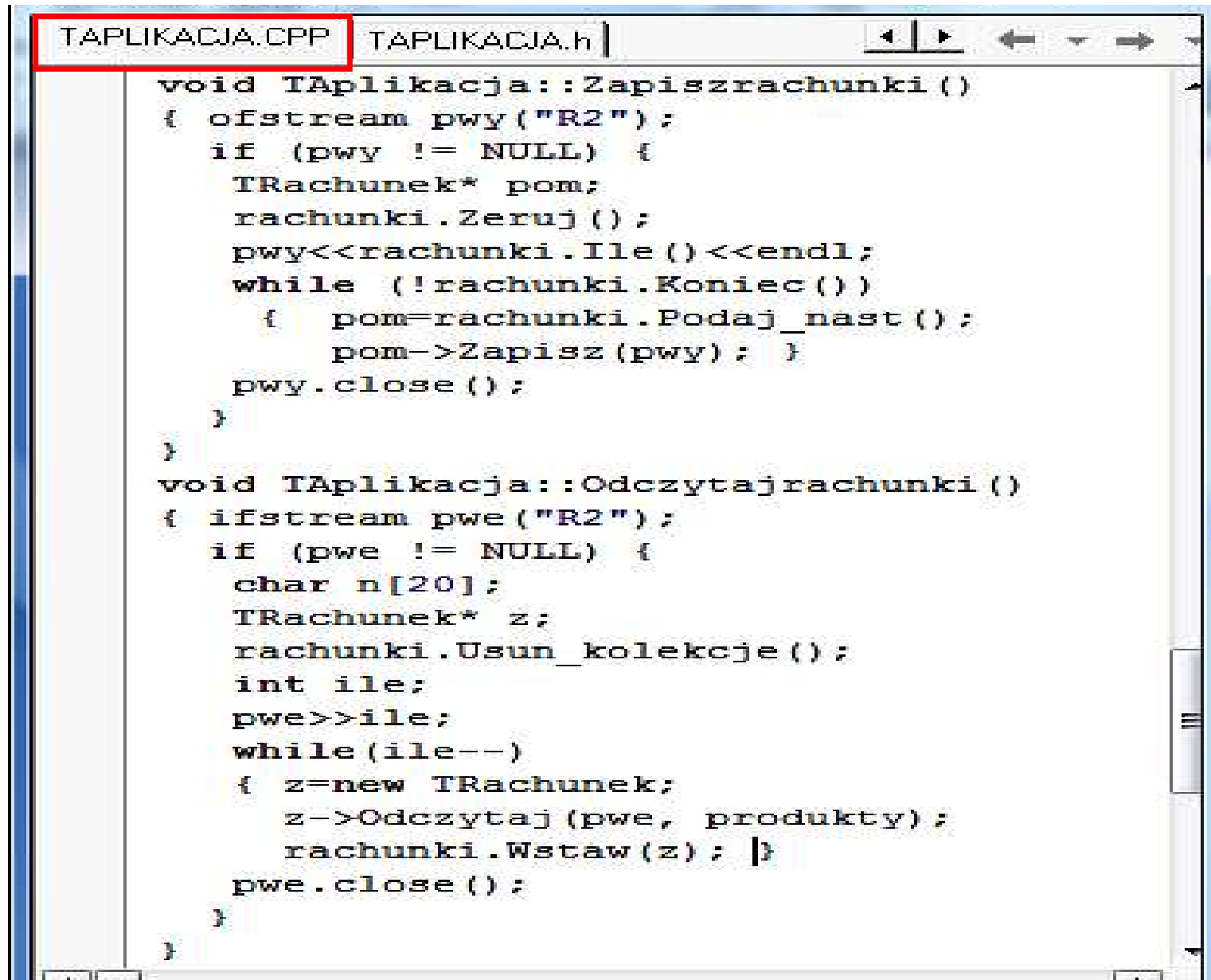
Build

Definicja metody **Odczytajprodukty** do odczytu kolekcji różnych typów produktów w pliku o nazwie „R1” w trybie tekstowym (przy odczycie obiekty są odtwarzane i zapisywane w pamięci) - **klasa TAplikacja**



```
TRACHUNEK.CPP | TRACHUNEK.H | TAPLIKACJA.CPP | TAPLIKACJA.H |
void TApplikacja::Odczytajprodukty()
{ ifstream pwe("R1");
  if (pwe !=NULL) {
    char n[20];
    TProdukt1* pom;
    int ilosc;
    produkty.Usun_kolekcje();
    pwe>>ilosc;
    while(ilosc--)
    { pwe>>n; //odczyt napisu <TProduktx>
      if (!strcmp(n, "<TProdukt1>"))      pom = new TProdukt1;
      else
        if (!strcmp(n, "<TProdukt2>"))      pom = new TProdukt2;
        else
          if (!strcmp(n, "<TProdukt3>"))      pom = new TProdukt3;
          else
            if (!strcmp(n, "<TProdukt4>"))      pom = new TProdukt4;
      if (pom)
        { pom->Odczytaj(pwe, produkty);
          produkty.Wstaw(pom); }
    }
    pwe.close();
  }
}
```

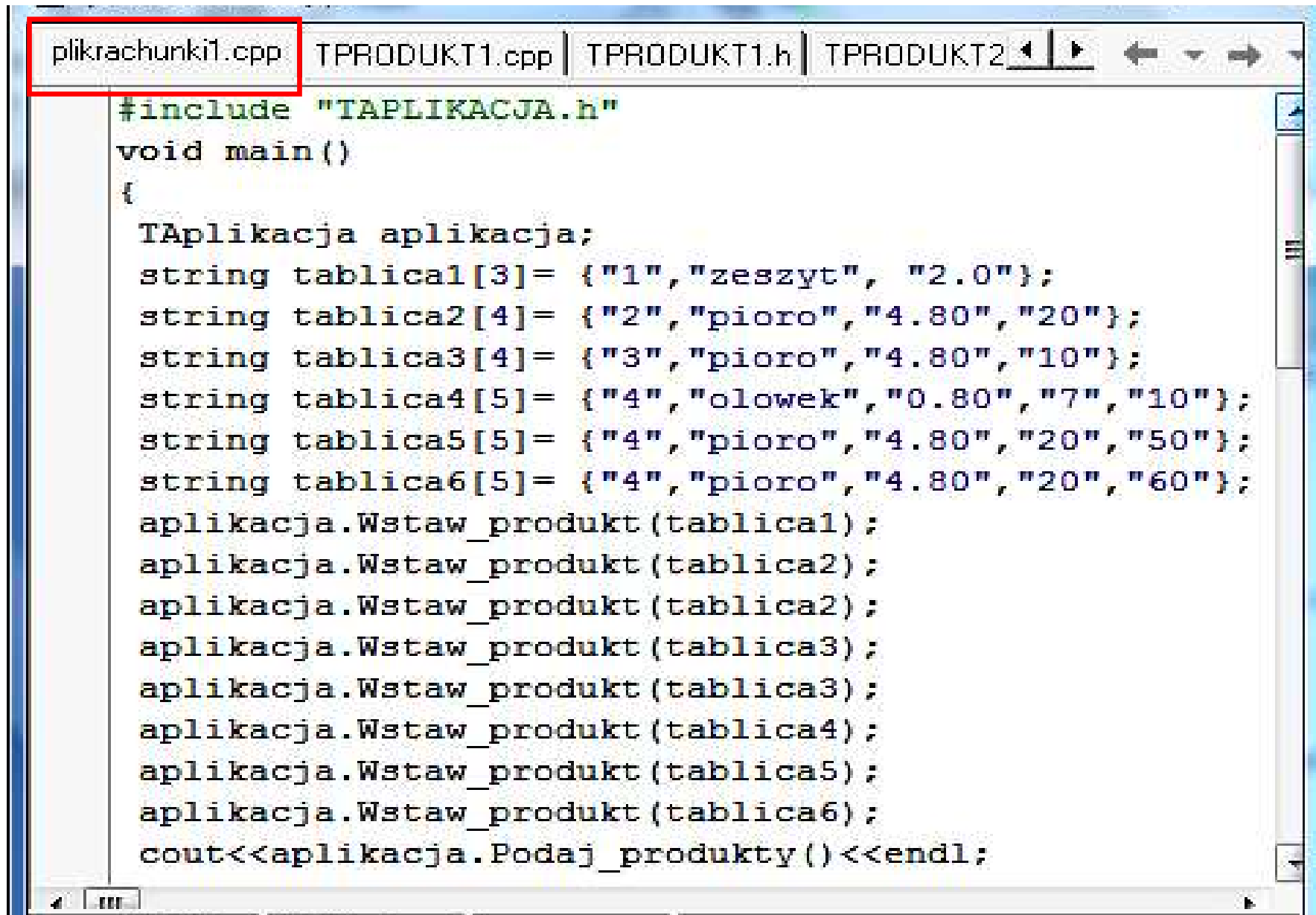
Definicja metody **Zapiszrachunki** do zapisu kolekcji rachunków oraz **Odczytajprodukty** do odczytu kolekcji rachunków z pliku o nazwie „R2” w trybie tekstowym (przy odczycie obiekty są odtwarzane i zapisywane w pamięci) - **klasa TApplikacja**



```
TAPLIKACJA.CPP | TAPLIKACJA.h |
void TApplikacja::Zapiszrachunki ()
{ ofstream pwy("R2");
  if (pwy != NULL) {
    TRachunek* pom;
    rachunki.Zeruj();
    pwy<<rachunki.Ile()<<endl;
    while (!rachunki.Koniec())
      { pom=rachunki.Podaj_nast();
        pom->Zapisz(pwy); }
    pwy.close();
  }
}

void TApplikacja::Odczytajrachunki ()
{ ifstream pwe("R2");
  if (pwe != NULL) {
    char n[20];
    TRachunek* z;
    rachunki.Usun_kolekcje();
    int ile;
    pwe>>ile;
    while (ile-->0)
      { z=new TRachunek;
        z->Odczytaj(pwe, produkty);
        rachunki.Wstaw(z); }
    pwe.close();
  }
}
```

Program demonstrujący zapis i odczyt obiektów przechowywanych w kolekcjach **produkty** i **rachunki** klasy typu **TAplikacja** – obiekty zapisane w formacie tekstowym przy odczycie są odtwarzane i zapisywane w pamięci.



```
plikrachunki1.cpp | TPRODUKT1.cpp | TPRODUKT1.h | TPRODUKT2
#include "TAPLIKACJA.h"
void main()
{
    TAplikacja aplikacja;
    string tablica1[3]= {"1", "zeszyt", "2.0"};
    string tablica2[4]= {"2", "pioro", "4.80", "20"};
    string tablica3[4]= {"3", "pioro", "4.80", "10"};
    string tablica4[5]= {"4", "olowek", "0.80", "7", "10"};
    string tablica5[5]= {"4", "pioro", "4.80", "20", "50"};
    string tablica6[5]= {"4", "pioro", "4.80", "20", "60"};
    aplikacja.Wstaw_produkt(tablica1);
    aplikacja.Wstaw_produkt(tablica2);
    aplikacja.Wstaw_produkt(tablica2);
    aplikacja.Wstaw_produkt(tablica3);
    aplikacja.Wstaw_produkt(tablica3);
    aplikacja.Wstaw_produkt(tablica4);
    aplikacja.Wstaw_produkt(tablica5);
    aplikacja.Wstaw_produkt(tablica6);
    cout<<aplikacja.Podaj_produkty()<<endl;
}
```

plikrachunki1.cpp

plikrachunki1.cpp

TPRODUKT1.cpp | TPRODUKT1

```
aplikacja.Wstaw_rachunek(1);  
aplikacja.Wstaw_rachunek(1);  
aplikacja.Wstaw_rachunek(2);  
cout<<aplikacja.Podaj_rachunki()<<endl;
```

```
aplikacja.Wstaw_zakup(tablica1,1,1);  
aplikacja.Wstaw_zakup(tablica1,2,1);  
aplikacja.Wstaw_zakup(tablica2,5,1);  
aplikacja.Wstaw_zakup(tablica2,6,1);  
aplikacja.Wstaw_zakup(tablica2,5,1);  
aplikacja.Wstaw_zakup(tablica3,6,1);  
aplikacja.Wstaw_zakup(tablica4,5,1);  
aplikacja.Wstaw_zakup(tablica4,6,1);  
aplikacja.Wstaw_zakup(tablica5,7,1);  
aplikacja.Wstaw_zakup(tablica6,7,1);
```

```
aplikacja.Wstaw_zakup(tablica1,1,2);  
aplikacja.Wstaw_zakup(tablica2,2,2);  
aplikacja.Wstaw_zakup(tablica3,6,2);  
aplikacja.Wstaw_zakup(tablica5,7,2);  
cout<<aplikacja.Podaj_rachunki()<<endl;
```

46: 55

Modified

Insert

Build


```
plikrachunki1.cpp | TPRODUKT1.cpp | TPRODUKT1.h | TPRODUKT2...
//zapis produktow i rachunkow
cout<<"Zapis produktow i rachunkow do plikow"<<endl;
aplikacja.Zapiszprodukty();
aplikacja.Zapiszrachunki();

cout<<"Odczyt produktow i rachunkow z plikow"<<endl;
aplikacja.Odczytajprodukty();
cout<<aplikacja.Podaj_produkty();
aplikacja.Odczytajrachunki();
cout<<aplikacja.Podaj_rachunki();

cin.get();
}
```

Po odczycie z pliku obiekty są pobrane z pojemników i są wyświetlane na ekranie

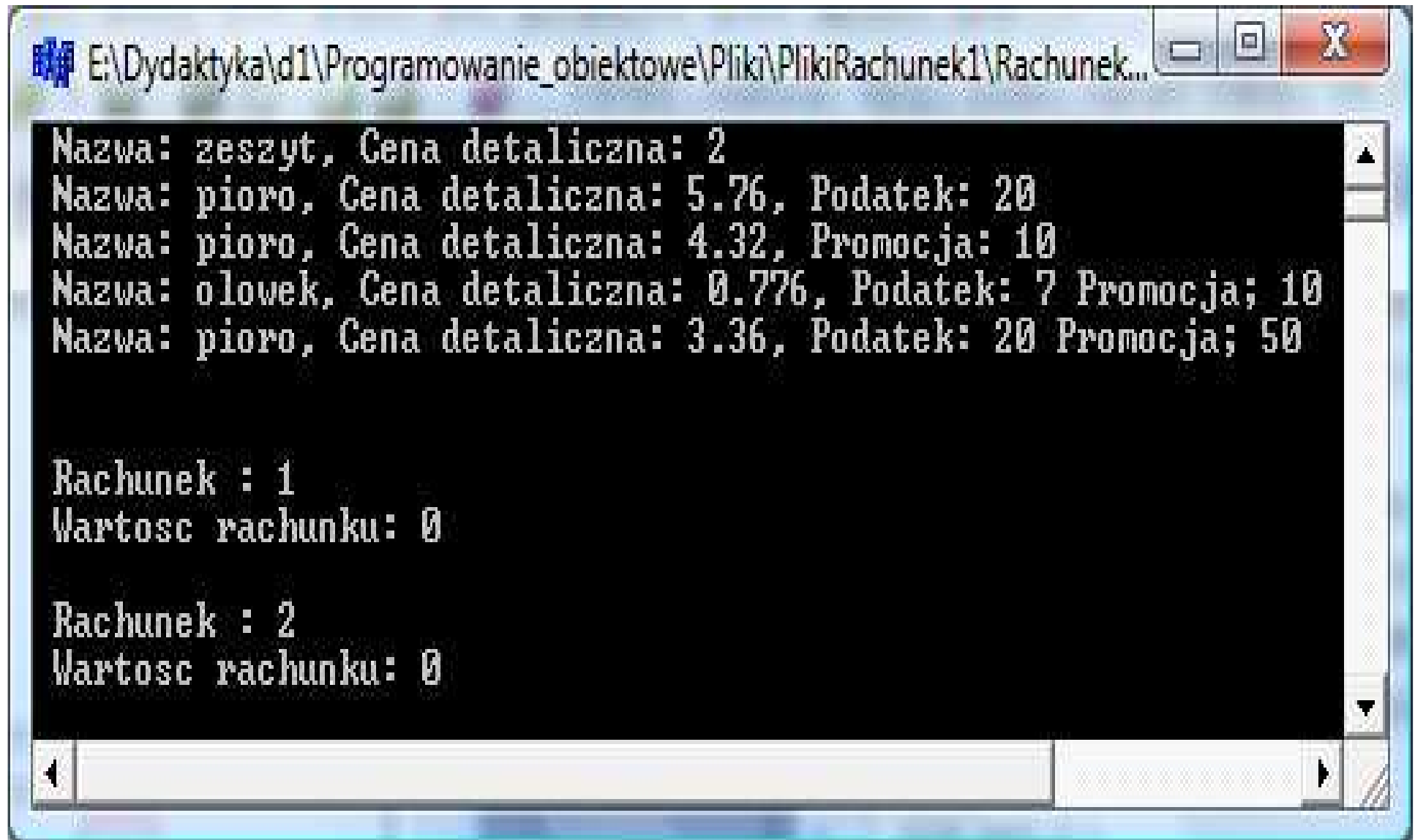
48: 46

Modified

Insert

Build

Program demonstrujący zapis i odczyt obiektów przechowywanych w kolekcjach **produkty** i **rachunki** klasy typu **TAplikacja** – obiekty zapisane w formacie tekstowym przy odczycie są odtwarzane i zapisywane w pamięci.



The screenshot shows a Windows command prompt window with the following text output:

```
E:\Dydaktyka\d1\Programowanie_obiektowe\Pliki\PlikiRachunek1\Rachunek...  
Nazwa: zeszyt, Cena detaliczna: 2  
Nazwa: pioro, Cena detaliczna: 5.76, Podatek: 20  
Nazwa: pioro, Cena detaliczna: 4.32, Promocja: 10  
Nazwa: olowek, Cena detaliczna: 0.776, Podatek: 7 Promocja; 10  
Nazwa: pioro, Cena detaliczna: 3.36, Podatek: 20 Promocja; 50  
  
Rachunek : 1  
Wartosc rachunku: 0  
  
Rachunek : 2  
Wartosc rachunku: 0
```

E:\Dydaktyka\d1\Programowanie_obiektowe\Pliki\PlikiRachunek1\Rachunek...

Rachunek : 1

Nazwa: zeszyt, Cena detaliczna: 2

Ilosc produktu: 3, Wartosc zakupu: 6

Nazwa: pioro, Cena detaliczna: 5.76, Podatek: 20

Ilosc produktu: 16, Wartosc zakupu: 92.2

Nazwa: pioro, Cena detaliczna: 4.32, Promocja: 10

Ilosc produktu: 6, Wartosc zakupu: 25.9

Nazwa: olowek, Cena detaliczna: 0.776, Podatek: 7 Promocja; 10

Ilosc produktu: 11, Wartosc zakupu: 8.54

Nazwa: pioro, Cena detaliczna: 3.36, Podatek: 20 Promocja; 50

Ilosc produktu: 7, Wartosc zakupu: 23.5

Wartosc rachunku: 156

Rachunek : 2

Nazwa: zeszyt, Cena detaliczna: 2

Ilosc produktu: 1, Wartosc zakupu: 2

Nazwa: pioro, Cena detaliczna: 5.76, Podatek: 20

Ilosc produktu: 2, Wartosc zakupu: 11.5

Nazwa: pioro, Cena detaliczna: 4.32, Promocja: 10

Ilosc produktu: 6, Wartosc zakupu: 25.9

Nazwa: pioro, Cena detaliczna: 3.36, Podatek: 20 Promocja; 50

Ilosc produktu: 7, Wartosc zakupu: 23.5

Wartosc rachunku: 63

E:\Dydaktyka\d1\Programowanie_obiektowe\Pliki\PlikiRachunek1\Rachunek...

Zapis produktow i rachunkow do plikow

Odczyt produktow i rachunkow z plikow

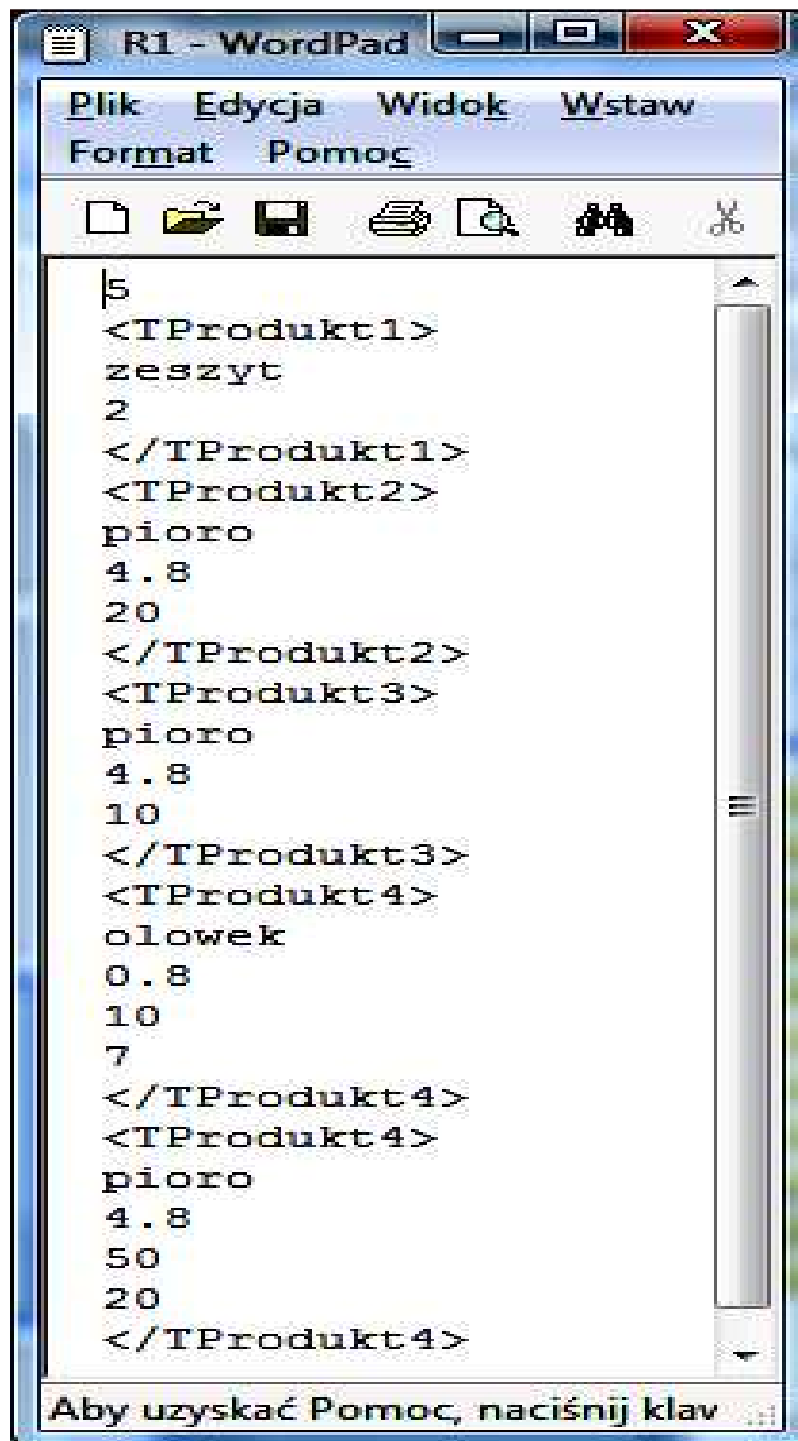
Nazwa: zeszyt, Cena detaliczna: 2
Nazwa: pioro, Cena detaliczna: 5.76, Podatek: 20
Nazwa: pioro, Cena detaliczna: 4.32, Promocja: 10
Nazwa: olowek, Cena detaliczna: 0.776, Podatek: 7 Promocja; 10
Nazwa: pioro, Cena detaliczna: 3.36, Podatek: 20 Promocja; 50

Rachunek : 1

Nazwa: zeszyt, Cena detaliczna: 2
Ilosc produktu: 3, Wartosc zakupu: 6
Nazwa: pioro, Cena detaliczna: 5.76, Podatek: 20
Ilosc produktu: 16, Wartosc zakupu: 92.2
Nazwa: pioro, Cena detaliczna: 4.32, Promocja: 10
Ilosc produktu: 6, Wartosc zakupu: 25.9
Nazwa: olowek, Cena detaliczna: 0.776, Podatek: 7 Promocja; 10
Ilosc produktu: 11, Wartosc zakupu: 8.54
Nazwa: pioro, Cena detaliczna: 3.36, Podatek: 20 Promocja; 50
Ilosc produktu: 7, Wartosc zakupu: 23.5
Wartosc rachunku: 156

Rachunek : 2

Nazwa: zeszyt, Cena detaliczna: 2
Ilosc produktu: 1, Wartosc zakupu: 2
Nazwa: pioro, Cena detaliczna: 5.76, Podatek: 20
Ilosc produktu: 2, Wartosc zakupu: 11.5
Nazwa: pioro, Cena detaliczna: 4.32, Promocja: 10
Ilosc produktu: 6, Wartosc zakupu: 25.9
Nazwa: pioro, Cena detaliczna: 3.36, Podatek: 20 Promocja; 50
Ilosc produktu: 7, Wartosc zakupu: 23.5
Wartosc rachunku: 63



The image shows a screenshot of a Windows WordPad application window titled "R1 - WordPad". The menu bar includes "Plik", "Edycja", "Widok", "Wstaw", "Format", and "Pomoc". The toolbar contains icons for file operations and editing. The main text area contains the following XML-like text:

```
|5
<TProdukt1>
zeszyt
2
</TProdukt1>
<TProdukt2>
pioro
4.8
20
</TProdukt2>
<TProdukt3>
pioro
4.8
10
</TProdukt3>
<TProdukt4>
olowek
0.8
10
7
</TProdukt4>
<TProdukt4>
pioro
4.8
50
20
</TProdukt4>
```

At the bottom of the window, there is a status bar that reads "Aby uzyskać Pomoc, naciśnij klav".

Zawartość pliku zawierającego zapis obiektów typów **TProdukt1**, **TProdukt2**, **TProdukt3**, **TProdukt4** w formacie tekstowym

