INFORMATYKA – MÓJ SPOSÓB NA POZNANIE I OPISANIE ŚWIATA

PROGRAM NAUCZANIA INFORMATYKI Z ELEMENTAMI PRZEDMIOTÓW MATEMATYCZNO-PRZYRODNICZYCH

Informatyka – poziom rozszerzony

Pomysł, przepis, program... i co dalej

Andrzej Ptasznik

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Tytuł: ***Pomysł, przepis, program... i co dalej***

Autor: ***Andrzej Ptasznik***

Redaktor merytoryczny: ***prof. dr hab. Maciej M. Sysło***

Materiał dydaktyczny opracowany w ramach projektu edukacyjnego
Informatyka – mój sposób na poznanie i opisanie świata.
Program nauczania informatyki z elementami przedmiotów matematyczno-przyrodniczych

www.info-plus.wwsi.edu.pl

infoplus@wwsi.edu.pl

Wydawca: Warszawska Wyższa Szkoła Informatyki
ul. Lewartowskiego 17, 00-169 Warszawa
www.wwsi.edu.pl
rektorat@wwsi.edu.pl

Projekt graficzny: *Marzena Kamasa*

Warszawa 2013

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

SCENARIUSZ TEMATYCZNY

POMYSŁ, PRZEPIS, PROGRAM... I CO DALEJ

INFORMATYKA – POZIOM ROZSZERZONY

OPRACOWANY W RAMACH PROJEKTU:

INFORMATYKA – MÓJ SPOSÓB NA POZNANIE I OPISANIE ŚWIATA.

PROGRAM NAUCZANIA INFORMATYKI

Z ELEMENTAMI PRZEDMIOTÓW MATEMATYCZNO-PRZYRODNICZYCH

Streszczenie

Problem wydawania reszty jest znanym zagadnieniem z dziedziny algorytmiki. Istotą problemu jest wybór zadanej kwoty przy użyciu minimalnej liczby nominałów z zadanego zbioru. Algorytm wydawania reszty jest wykorzystywany w wielu urządzeniach, które muszą wydawać określoną kwotę (bankomaty, automaty do napojów itp.). W ramach scenariusza zaproponowane zostanie rozwiązanie algorytmu wydawania reszty w podejściu obiektowym z zastosowaniem rekurencyjnego konstruktora. Kolejnym krokiem będzie napisanie prostego programu symulującego automat wydający określoną kwotę.

Czas realizacji

2 x 45 minut

Tematy lekcji

1. Problem i jego rozwiązanie
2. Wykorzystanie algorytmu

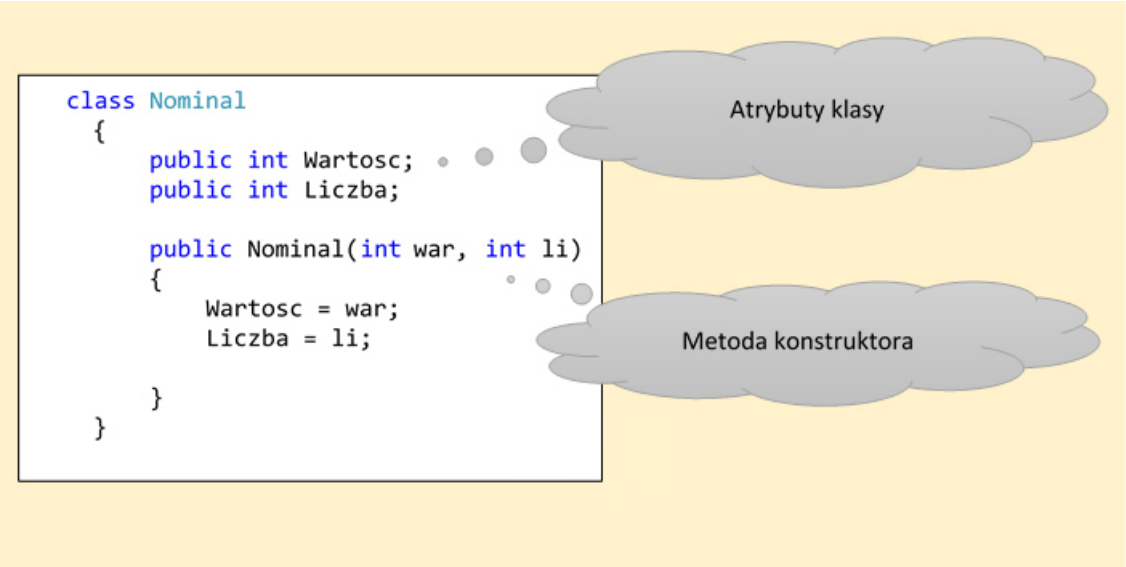
LEKCJA NR 1

TEMAT: Algorytm i jego rozwiązanie

Streszczenie

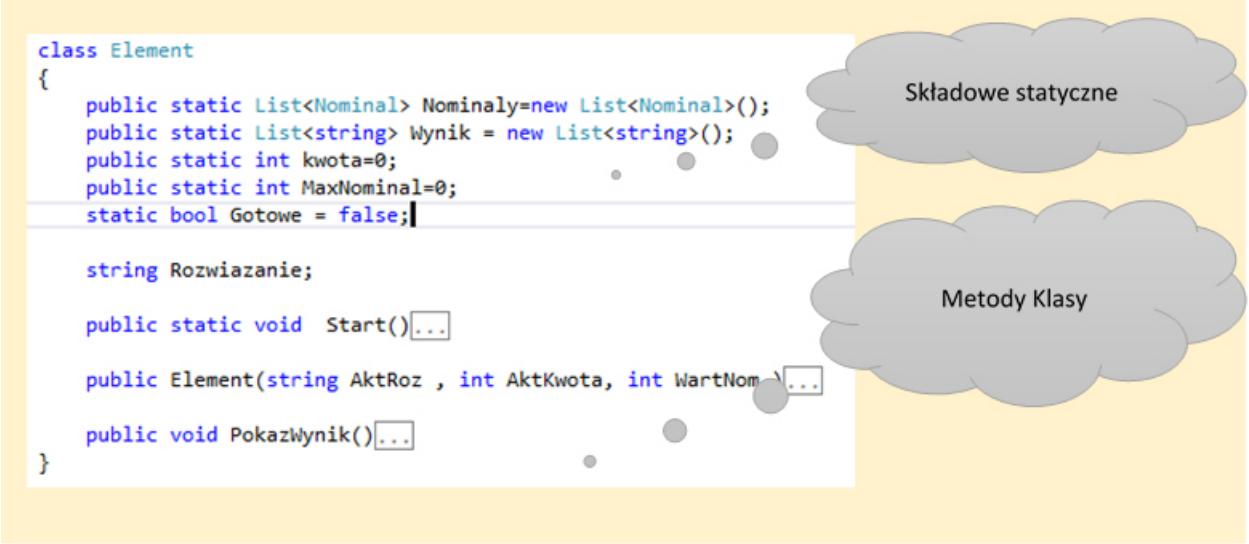
Algorytm wydawania reszty jest znanym problemem algorytmicznym. Zwykle przedstawiany jest jako problem wydania określonej kwoty za pomocą najmniejszej liczby dostępnych w nieograniczonej ilości nominałów. Algorytm wydawania reszty jest wykorzystywany w praktyce w urządzeniach wypłacających określone kwoty, jak bankomaty i automaty do zakupów. We wspomnianych urządzeniach nie można stosować założenia, że liczba walorów danego nominału jest nieograniczona, gdyż każda zrealizowana transakcja zmienia dostępny ich zbiór.

W ramach scenariusza omówiony zostanie przykład rozwiązania algorytmu wydawania reszty z założeniem, że dysponujemy określoną liczbą poszczególnych nominałów. Proponowane rozwiązanie jest dość nietypowe, gdyż do jego rozwiązania zastosujemy programowanie obiektowe z wykorzystaniem rekurencyjnego konstruktora klasy. Propozycja obiektowego podejścia do problemu wynika z faktu, że w kolejnym kroku będziemy wykorzystywać zdefiniowaną klasę do realizacji programu symulującego działanie urządzenia, którego jednym z zadań jest wydawanie zadanej kwoty. Na rysunku 1 pokazano fragment kodu w języku C#, w którym deklarujemy klasę o nazwie Nominal.



Rysunek 1. Deklaracja klasy o nazwie Nominal.

Deklaracja klasy Nominal jest bardzo prosta i składa się z dwóch atrybutów oraz metody konstruktora. Atrybut o nazwie Wartosc określa wartość nominału, a atrybut o nazwie Liczba określa liczbę dostępnych egzemplarzy danego nominału. Na rysunku 2 pokazano deklarację klasy o nazwie Element, która jest podstawą proponowanego rozwiązania.

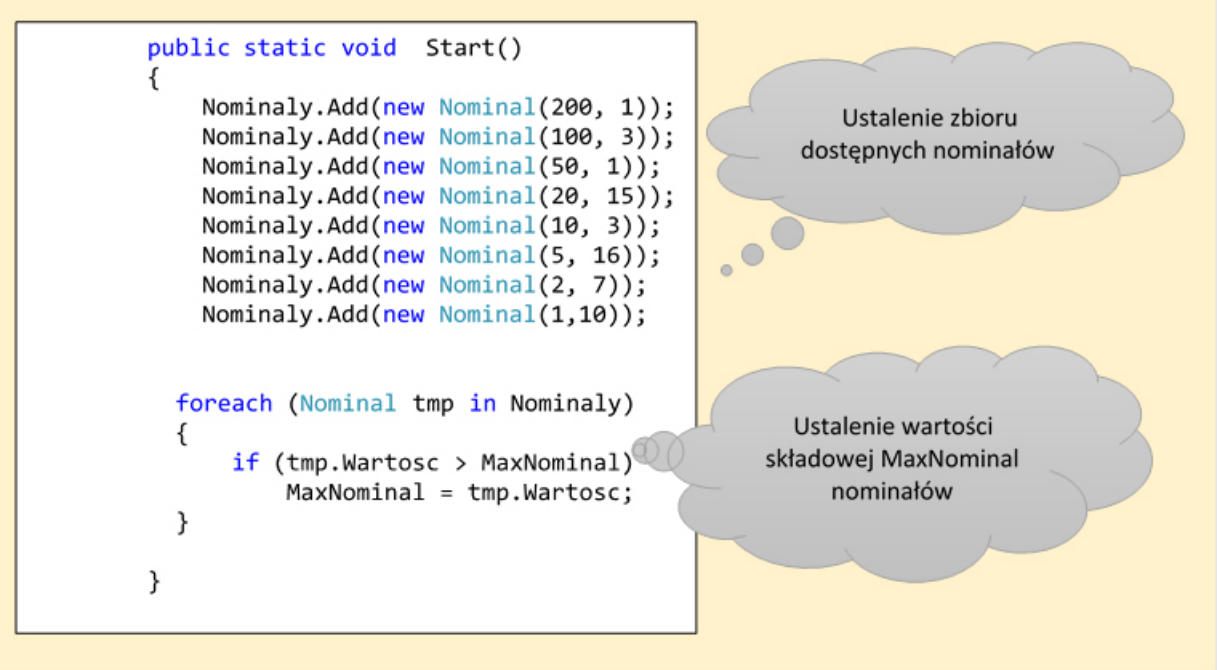


Rysunek 2. Deklaracja klasy o nazwie Element.

Opis składowych klasy:

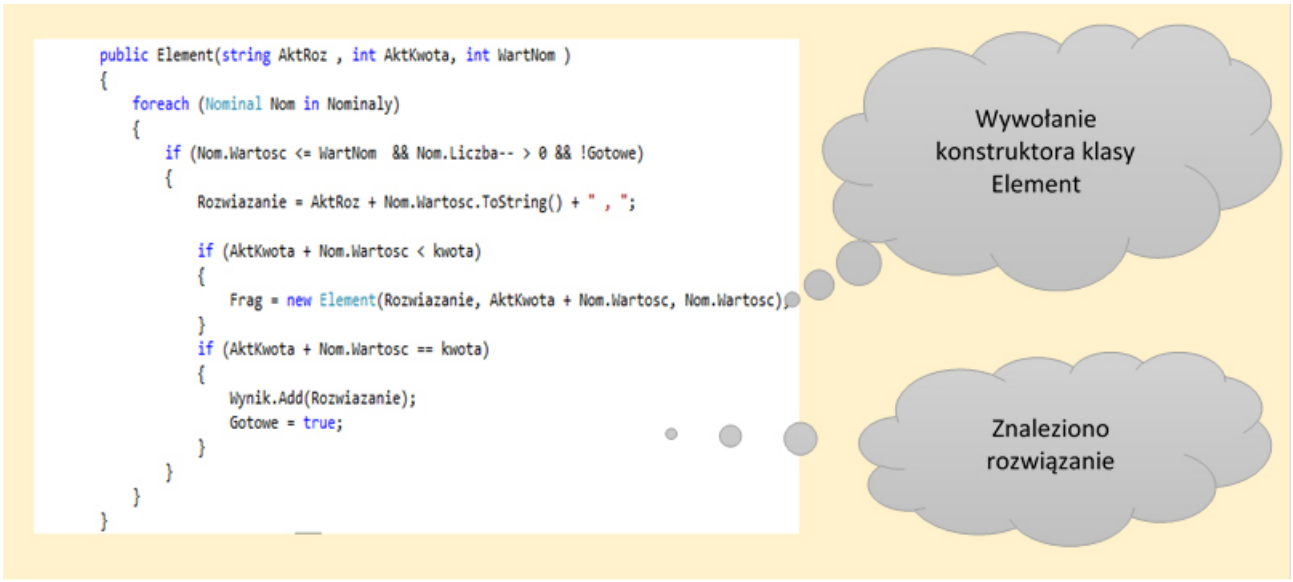
- public static List<Nominal> Nominaly=new List<Nominal>(); – deklaracja statycznej listy obiektów klasy Nominal, która reprezentuje dostępny zbiór nominałów,
- public static List<string> Wynik = new List<string>(); – deklaracja statycznej listy przechowującej elementy typu string, w programie będzie przechowywać odnalezione rozwiązania,
- public static int kwota=0; – statyczna składowa reprezentująca poszukiwana kwotę reszty,
- public static int MaxNominal=0; – statyczna składowa reprezentująca wartość maksymalnego nominału dostępnego w liście Nominaly,
- static bool Gotowe = false; – statyczna składowa określająca status wyszukiwania,
- public static void Start(){ } – statyczna metoda klasy, której zadaniem jest zainicjowanie listy Nominaly i ustalenie wartości MaxNominal,
- string Rozwiazanie; – składowa zawierająca aktualną postać rozwiązania,
- public Element(string AktRoz, int AktKwota, int WartNom {} – konstruktor klasy,
- public void PokazWynik() {} – metoda prezentująca wynik rozwiązania.

Najważniejszą składową w ukazanej na rysunku 2 deklaracji klasy jest metoda konstruktora, bo w niej jest zawarte rozwiązanie problemu wydawania wskazanej kwoty. Na rysunku 3 pokazano kod metody klasy Element o nazwie Start. Zadaniem metody jest zainicjowanie listy Nominaly oraz określenie wartości składowej klasy o nazwie MaxNominal. W zadaniach dla uczniów można założyć rozbudowę tej metody o inne formy ustalenia zbioru dostępnych nominałów, takie jak wczytanie danych z pliku lub wczytywanie z klawiatury wartości wpisywanych przez użytkownika.



Rysunek 3. Deklaracja metody Start.

Na rysunku 4 widać kod deklaracji konstruktora klasy Element, zawierające rozwiązanie algorytmu wydawania reszty.



Rysunek 4. Kod konstruktora klasy Element.

Konstruktor klasy Element jest wywoływany z trzema parametrami:

- AktRoz – parametr, w którym przekazywana jest aktualna postać rozwiązania.
- AktKwota – parametr, który reprezentuje kwotę, która w danym momencie pozostała do wyszukania.
- WartNom – wartość nominału, który w danym momencie rozpatrujemy jako element rozwiązania.

Utworzenie w programie obiektu klasy Element powoduje rozwiązanie problemu i zapisanie wyniku w liście Wynik.

Na rysunku 5 zaprezentowano kod metody PokazWynik, której zadaniem jest wyświetlenie na ekranie znalezionego rozwiązania.

```
        public void PokazWynik()
        {
            Console.WriteLine();
            if (Element.Gotowe==false)
            {
                Console.WriteLine("Niestety - brak rozwiązania");
            }
            else
            {
                foreach (string tmp in Element.Wynik)
                {
                    Console.WriteLine(tmp);

                }
            }
            Console.ReadLine();
        }
    }
```

Gdyby nie przerywano działania po znalezieniu pierwszego rozwiązania, wyświetlony zostałby zbiór rozwiązań

Rysunek 5. Kod metody PokazWynik.

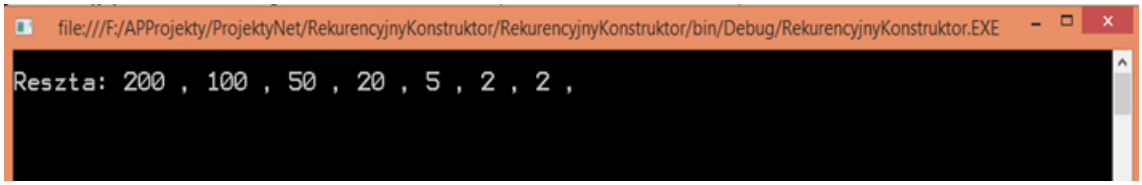
Zaproponowane rozwiązanie algorytmu wydawania reszty można w prosty sposób zmodyfikować, by osiągnąć dodatkowe efekty jego działania. Mogą nimi być: wyszukanie wszystkich możliwych kombinacji dostępnych nominałów dających w efekcie poszukiwaną kwotę lub korygowanie liczby poszczególnych nominałów po znalezieniu rozwiązania w taki sposób, żeby kolejne wywołanie konstruktora klasy Element uwzględniało zmianę zbioru nominałów wynikającą z odnalezionego wcześniej rozwiązania. Na rysunku 6 pokazano postać funkcji Main. W pokazanym kodzie ustalono wyszukiwaną kwotę (Element. kwota=379) i wywołano statyczną metodę Start klasy Element, która inicjuje zbiór nominałów oraz ustala maksymalną wartość nominału w zbiorze. Rozwiązaniem problemu jest utworzenie obiektu klasy Element, któremu przekazywane są inicjujące wartości parametrów.

```
class Program
{
    static void Main(string[] args)
    {
        Element.kwota = 379;
        Element.Start();

        Element wyn = new Element("Reszta: ",0,Element.MaxNominal);
        wyn.PokazWynik();

    }
}
```

Rysunek 6. Kod funkcji Main.

Utworzenie obiektu klasy Element jest równoznaczne ze znalezieniem rozwiązania i dlatego ostatnim krokiem jest wywołanie metody PokazWynik.

Na rysunku 7 uwidoczniono wyświetloną na ekranie postać wyniku dla kwoty 379.



Rysunek 7. Wynik działania metody Main dla kwoty 379.

W sytuacji, gdy dostępny zbiór nominałów nie daje żadnego rozwiązania, wyświetlany jest wynik działania pokazany na rysunku 8.

Rysunek 8. Wynik działania programu przy braku rozwiązania.

Szczegóły zaproponowanego rozwiązania omówione są w filmie instruktażowym o nazwie Algorytm i rozwiązanie.

Podstawa programowa

Etap edukacyjny: IV, przedmiot: informatyka (poziom rozszerzony)

Cele kształcenia – wymagania ogólne

III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.

Treści nauczania – wymagania szczegółowe

Uczeń:

1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin;
2) stosuje podejście algorytmiczne do rozwiązywania problemu;
3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;
4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;
5) posługuje się podstawowymi technikami algorytmicznymi;
6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;
7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowa nie rozwiązania.

Cel

Podstawowym celem lekcji jest opracowanie algorytmu wydawania reszty w podejściu obiektowym. Rozwiązanie to zostanie wykorzystane w kolejnej lekcji do tworzenia programu symulującego działanie urządzenia wydającego określoną kwotę.

Słowa kluczowe

klasa, konstruktor, rekurencja, algorytm

Co przygotować

- Zainstalować Microsoft Visual Studio Express 2012 for Windows Desktop. Oprogramowanie, które jest darmową wersją pakietu Visual Studio 2012, można pobrać pod adresem internetowym:

  http://www.microsoft.com/en-us/download/details.aspx?id=40787

- Nauczyciel może wykorzystać inne środowisko programistyczne, o ile jest mu dobrze znane. Wprowadzenie do cyklu lekcji scenariusza warto omówić z wykorzystaniem prezentacji, zawartej w plikach dodatkowych, o nazwie „Wydawanie reszty – rozwiązanie i wykorzystanie"

Przebieg zajęć

Wprowadzenie (15 minut)

W trakcie wprowadzenia wykorzystujemy prezentację o nazwie „Wydawanie reszty – rozwiązanie i wykorzystanie".

1. Na początku zajęć nauczyciel podaje temat lekcji „Algorytm i jego rozwiązanie" i krótko omawia podstawowe problemy związane z algorytmem wydawania reszty.
2. Po omówieniu podstaw, nauczyciel przystępuje do zaprezentowania uczniom omówionego w streszczeniu lekcji rozwiązania i przedstawia dalsze kierunki prac związanych z jego wykorzystaniem.
3. Na zakończenie tej części lekcji prezentuje sposób uruchomienia środowiska Visual Studio 2012 i przystępuje do realizacji zadania. Szczegóły zadania opisane zostały w streszczeniu do lekcji.

Zasadnicza część lekcji (25 minut)

W zasadniczej części lekcji, wspólnie z uczniami, omawiane i realizowane jest wykonanie programu i jego testowanie. Omówić należy możliwe kierunki modyfikacji zaproponowanego algorytmu i zaproponować konkurs na wykonanie najciekawszych jego rozszerzeń.

Podsumowanie (5 minut)

W ramach dyskusji należy zwrócić szczególną uwagę na fakt, że zdefiniowana klasa będąca rozwiązaniem algorytmu będzie wykorzystana przy tworzeniu programu symulującego działanie automatu wydającego resztę.

Sprawdzenie wiedzy

Ocenianie

Podstawowe sprawdzenie wiedzy można zrealizować na podstawie aktywności uczniów w trakcie omawiania zadania oraz oceny wykonanych modyfikacji algorytmu.

Dostępne pliki

1. Prezentacja wprowadzająca
2. Film instruktażowy 1 – Algorytm i rozwiązanie
3. Plik skompresowany RekurencyjnyKonstruktor.zip zawierający folder omawianego projektu zrealizowanego w środowisku Visual Studio Express 2012 for Windows Desktop (materiały pomocnicze 1)

LEKCJA NR 2

TEMAT: Wykorzystanie algorytmu

Streszczenie

W ramach drugiej lekcji zostanie zaprezentowane wykorzystanie algorytmu wydawania reszty w programie, który będzie symulował działanie pewnego urządzenia. Program zostanie wykonany w środowisku Microsoft Visual Studio Express 2012 for Windows Desktop. Jednym z zadań programu będzie graficzna prezentacja wydanej reszty i w tym celu zostanie zdefiniowana klasa o nazwie Obraz. Obiekty tej klasy będą pokazywały graficzną postać wybranych nominałów. Na rysunku 9 pokazano deklarację klasy Obraz, a szczegóły techniczne związane z obsługą plików graficznych omówiono w filmie instruktażowym „Wykorzystanie algorytmu".


```
public Obraz(int war)
        {
            switch (war)
            {
                case 1: BackgroundImage = Properties.Resources._1PLN;
                    Width = 80;
                    Height = 80;
                    break;
                case 2: BackgroundImage = Properties.Resources._2PLN;
                    Width = 80;
                    Height = 80;
                    break;
                case 5: BackgroundImage = Properties.Resources._5PLN;
                    Width = 80;
                    Height = 80;
                    break;
                case 10: BackgroundImage = Properties.Resources._10PLN;
                    Width = 160;
                    Height = 80;
                    break;
                case 20: BackgroundImage = Properties.Resources._20PLN;
                    Width = 160;
                    Height = 80;
                    break;
                case 50: BackgroundImage = Properties.Resources._50PLN;
                    Width = 160;
                    Height = 80;
                    break;
                case 100: BackgroundImage = Properties.Resources._100PLN;
                    Width = 160;
                    Height = 80;
                    break;
                case 200: BackgroundImage = Properties.Resources._200PLN;
                    Width = 160;
                    Height = 80;
                    break;
            }
            BackgroundImageLayout = ImageLayout.Stretch;

        }
```

Rysunek 9. Deklaracja klasy Obraz.

Podstawowym zadaniem jest wykorzystanie algorytmu wykonanego podczas pierwszej lekcji scenariusza, w którym należy wprowadzić kilka drobnych modyfikacji.

Na rysunku 10 pokazano deklaracje klasy Element i łatwo zauważyć, że jedyną zmianą w stosunku do rozwiązania pokazanego na rysunku 2 jest brak elementu Rozwiazanie oraz inny typ danych zapisywanych w liście Wynik. Zmiany wynikają z konieczności dostosowania rozwiązania do graficznej prezentacji wyniku.

```
class Element
{
    public static List<Nominal> Nominaly=new List<Nominal>();
    public static int kwota=0;
    public static int MaxNominal=0;
    public static List<Obraz> Wynik = new List<Obraz>();
    public static bool Gotowe = false;
    Element Frag;


    public static void  Start()...

    public Element(int AktKwota, int WartNom)...

    public void PokazWynik()...

}
```

Rysunek 10. Deklaracja klasy Element.

Podstawą rozwiązania jest konstruktor klasy Element, w którym zastosowano rekurencyjne rozwiązanie problemu wydawania reszty. Na rysunku 11 pokazano postać deklaracji konstruktora klasy Element. W porównaniu z pokazanym na rysunku 4 konstruktorem wprowadzone zostały dwie zmiany związane z inną formą prezentacji wyniku: brak parametru AktRoz oraz obsługa zapisywania obiektów obraz do listy Wynik.

```
public Element(int AktKwota, int WartNom)
{

    foreach (Nominal Nom in Nominaly)
    {

        if (Nom.Wartosc <= WartNom  && Nom.Liczba-- > 0 && !Gotowe)
        {
            if (AktKwota + Nom.Wartosc < kwota)
            {
                Wynik.Add(new Obraz(Nom.Wartosc));
                Frag = new Element(  AktKwota + Nom.Wartosc, Nom.Wartosc);
            }
            if (AktKwota + Nom.Wartosc == kwota)
            {
                Wynik.Add(new Obraz(Nom.Wartosc));
                Gotowe = true;
            }
        }
    }
}
```

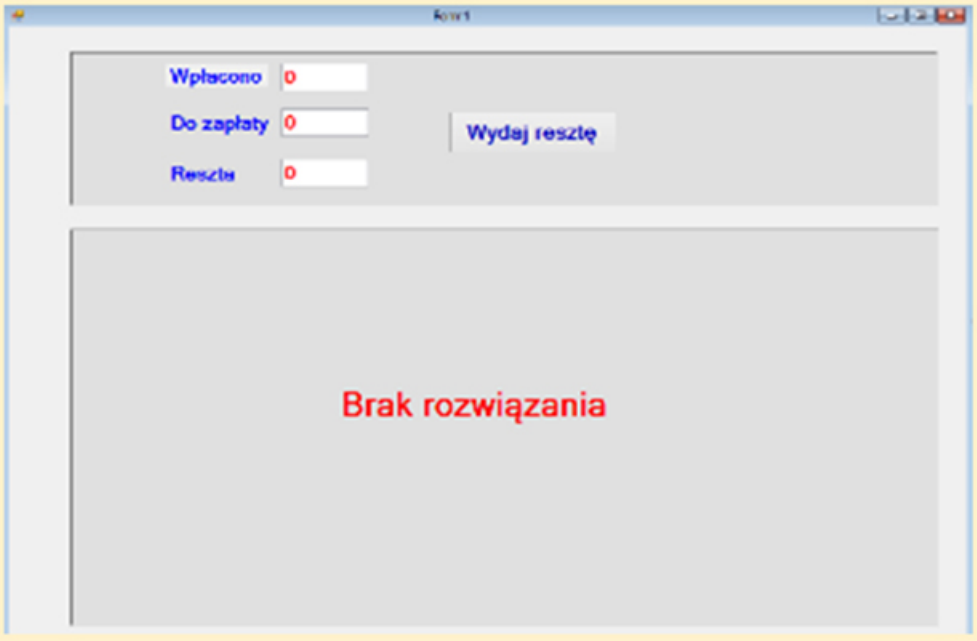
Rysunek 11. Deklaracja metody konstruktora klasy Element.

Pokazane na rysunkach 10 i 11 deklaracje niewiele różnią się od tych samych elementów zrealizowanych w ramach lekcji pierwszej scenariusza. Korekta musiała zostać przeprowadzona w metodzie PokazWynik, co ukazuje rysunek 12.

```
public void PokazWynik()
{
    int i = 0;
    if (Gotowe==true)
    {
        foreach (Obraz tmp in Wynik)
        {
            i++;
            tmp.Parent = Form1.FormRef.PanelReszta;
            switch (i % 3)
            {
                case 0: tmp.Left = 50; break;
                case 1: tmp.Left = 250; break;
                case 2: tmp.Left = 450; break;
            }

            tmp.Top = 85 * ((i / 3) ) + 5;
        }
    } else
        {
            Form1.FormRef.Komunikat.Visible=true;
        }
}
```

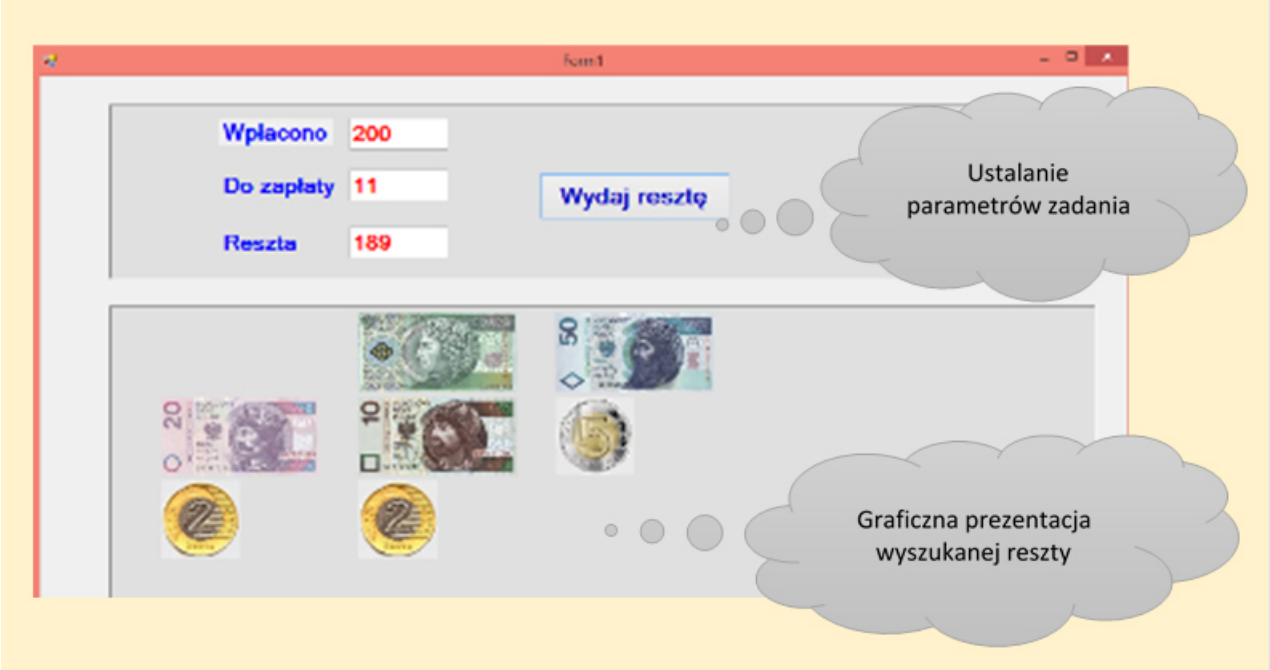
Rysunek 12. Deklaracja metody PokazWynik klasy Element.

Istotą metody PokazWynik jest odpowiednie umieszczenie obrazów nominałów w oknie aplikacji. W kolejnym kroku przygotowane zostanie okno, w którym znajdą się pola do wprowadzenia Kwoty, jaką użytkownik wpłaca, pola, w których należy określić kwotę do zapłaty i pola wyświetlające resztę do wydania. Przycisk oznaczony etykietą Pokaz wynik uruchamia rozwiązanie i wyświetlenie wydawanej reszty. Na rysunku 12 widzimy postać okna, w którym będziemy testowali rozwiązanie problemu.



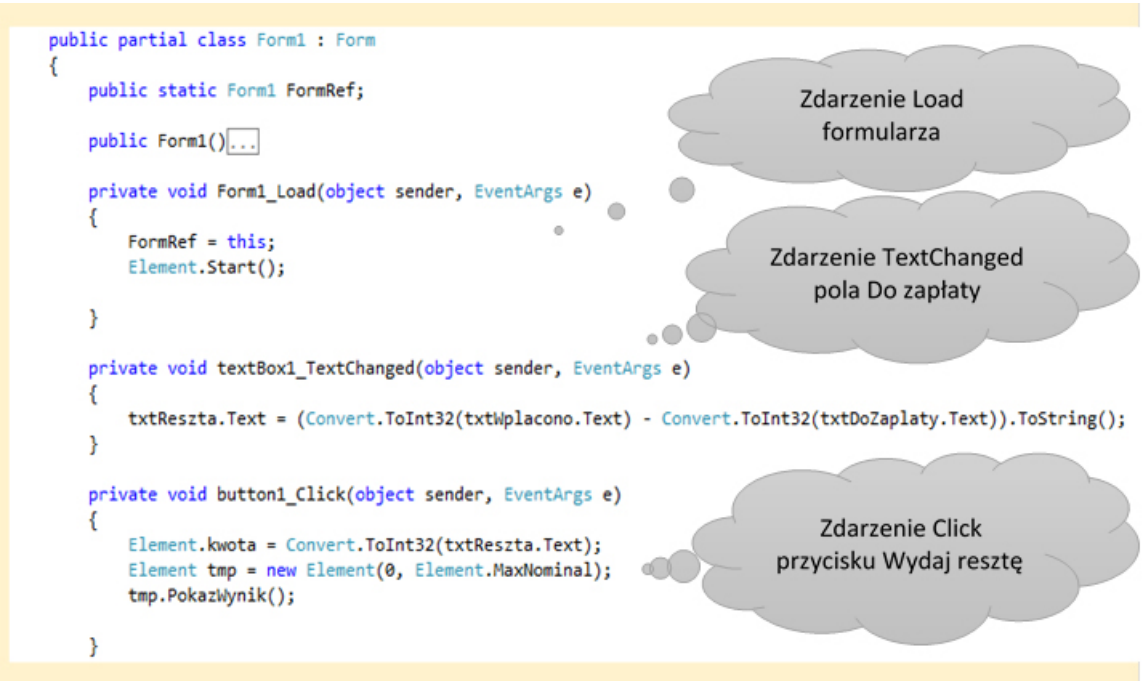
Rysunek 13. Postać okna do testowania działania algorytmu.

Działanie programu polega na wpisaniu odpowiednich kwot do pól Wpłacono i Do zapłaty, co spowoduje automatyczne ustalenie kwoty wydawanej reszty. Naciśnięcie przycisku Wydaj resztę spowoduje uruchomienie rozwiązania algorytmu i zaprezentowanie wyniku. Na rysunku 14 pokazano postać okna po uruchomieniu rozwiązania dla przykładowych kwot.



Rysunek 14. Postać okna prezentującego wynik działania algorytmu.

Działanie programu wymagało napisania krótkich fragmentów kodu będących obsługą zdarzeń. Wykorzystywane są trzy zdarzenia, zdarzenie Load formularza, zdarzenie TextChanged pola Do zapłaty oraz zdarzenie Click przycisku Wydaj resztę. Na rysunku 15 pokazano kod wymienionych metod obsługi zdarzeń.



Rysunek 15. Kod metod obsługi zdarzeń.

W zdarzeniu Load formularza wykonywana jest metoda Start klasy Element, która inicjuje zbiór nominałów oraz zapamiętywana jest referencja do tworzonego obiektu formularza w zmiennej FormRef. Zdarzenie generowane jest w momencie uruchamiania formularza i czynności w nim wykonywane przygotowują program do realizacji przewidzianych zadań. W zdarzeniu TextChanged pola Do zapłaty ustalana jest wartość wyświetlana w polu Reszta. Podstawowe działanie realizowane jest w zdarzeniu Click przycisku Wydaj resztę, a podstawą działania jest tworzenie obiektu klasy Element oraz wykonanie metody PokazWynik, która umieszcza w oknie graficzne obrazy nominałów składające się na wynik rozwiązania. Na rysunku 16 pokazano postać okna w sytuacji, gdy dla dostępnego zbioru nominałów nie istnieje rozwiązanie.

Rysunek 16. Postać okna przy braku rozwiązania.

Omówiony przykład programu powinien być wstępem do przydzielenia uczniom zadań związanych z jego modyfikacją. Propozycje modyfikacji programu zostały omówione w dodatkowym pliku o nazwie Zadania.

Podstawa programowa

Etap edukacyjny: IV, przedmiot: informatyka (poziom rozszerzony)

Cele kształcenia – wymagania ogólne

III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.

Treści nauczania – wymagania szczegółowe

Uczeń:

1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin;
2) stosuje podejście algorytmiczne do rozwiązywania problemu;
3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera;
4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji;
5) posługuje się podstawowymi technikami algorytmicznymi;

6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania;
7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania.

Cel

Podstawowym celem lekcji jest kontynuacja zadania, którego przygotowanie rozpoczęto w ramach poprzedniej lekcji scenariusza. Głównym celem jest napisanie programu symulującego działanie urządzenia wydającego resztę, w którym wykorzystany zostanie algorytm opracowany i omówiony w ramach pierwszej lekcji scenariusza.

Słowa kluczowe

funkcja, parametry funkcji, biblioteka funkcji

Co przygotować

- W ramach lekcji wykorzystywany jest szkielet programu, który został przygotowany w trakcie lekcji pierwszej oraz deklaracje funkcji przygotowane przez zespoły uczniów.

Przebieg zajęć

Wprowadzenie (10 minut)

W trakcie wprowadzenia należy przypomnieć i pokazać program, który wykonany został w ramach lekcji pierwszej. Omówić należy istotę zadania, czyli deklarację klasy Element i istotę rozwiązania problemu wydawania reszty, w którym wykorzystaliśmy rekurencję w metodzie konstruktora klasy.

Zasadnicza część lekcji (30 minut)

Wspólnie z uczniami omawiamy i wykonujemy prosty przykład programu prezentującego wydaną resztę jako graficzne obrazy nominałów.

Dyskusja podsumowująca (10 minut)

W ramach dyskusji należy ocenić wykonany program, ocenić poszczególne etapy rozwiązania oraz omówić możliwości dalszych modyfikacji.

Sprawdzenie wiedzy

W celu sprawdzenia wiedzy należy przydzielić uczniom zadanie polegające na rozszerzeniu funkcjonalności zaprezentowanego programu.

Ocenianie

Ocena uczniów na podstawie aktywności w trakcie lekcji oraz na podstawie oceny wykonania zadania.

Dostępne pliki

1. Film instruktażowy 2 – Wykorzystanie algorytmu
2. Plik skompresowany ResztaWindows.zip zawierający folder omawianego projektu zrealizowanego w środowisku Visual Studio Express 2012 for Windows Desktop (materiały pomocnicze 2)
3. Zadania
4. Test

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska
Wyższa Szkoła
Informatyki

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego