

2020/21 - Wprowadzenie do programowania (C#) - LAB (K. Molenda)

Kokpit / Moje kursy / wpd-2020-21-lab-kmolenda / Lab. 09-10. OOP, interfejsy / Zadanie: klasa Person, klasa Child, dziedziczenie

Rozpoczęto	Thursday, 3 June 2021, 12:59 PM
Stan	Ukończono
Ukończono	Friday, 11 June 2021, 22:26 PM
Wykorzystany czas	1 dni 9 godzin

Ocena: 5,00 pkt. na 5,00 pkt. możliwych do uzyskania (100%)

Pytanie 1	Poprawne
Ocena:	5,00
% Odpowiedzi	pytanie

Jesteli poproszony o zamodelowanie klas dla aplikacji przechowujace dane osob. Musisz przewidziec, ze bedzesz przechowywal informacje na temat osob doroslych i dzieci. Dzieci dziedzicza pewne cechy od swoich rodzicow.

Tworzysz:

- Klasę `Person` – będąca klasą bazową z której pozoistale klasa dziedziczą.
- Osoba ma imię (`firstName`) i nazwisko (`familyName`).
- Imię i nazwisko sa pojedynczymi wyrazami, nie zawierajac spacji, zaczynaja sie od duzej litery, zaś pozostałe sa literami malymi. Stosowne korekty wprowadz w chwili przypisywania wartosci (usun spacje, skoryguj format "duza litera-male litera"). Jesli napis mialby zawierac znaki, które nie sa literami, lub byliby pusty zglosz wyjątek `ArgumentException` z komunikatem "Wrong name!".
- wiek osoby nie moze byc ujemny (zglosz wyjątek `ArgumentException` z komunikatem "Age must be positive!").
- Zdefiniuj konstruktor, akceptujacy parametry (`familyName` oraz `age` w klasie `Person`), publiczne do odczytu i chronione do zapisu (dostepne tylko z poziomu klas i klas dziedziczących).
- Zdefiniuj konstruktor reprezentacyjny obiektu, zwierajaca napisem (`ToString`) (`Object`)
- Zdefiniuj metody modyfikujace imię, nazwisko oraz wiek, o dostepie publicznym i o nazwach: `modifyFirstName`, `modifyFamilyName`, `modifyAge`.
- Zawiera referencje do rodziców matki (`mother`) oraz ojca (`father`). Referencje te ustawiane sa w momencie konstrukcji obiektu i nie mogą byc już nigdy zmienione. Jesli nie są znani rodzice dziecka, przypisz referencjom `null`.
- Dziecko nie moze miec wieku wiekszego niz 15 lat (zglosz wyjątek `ArgumentException` z komunikatem "Child's age must be less than 15!").
- Modyfikacja wieku dziecka moziala jedynie w zakresie od 0 do 15 lat.
- Tekstowa reprezentacja obiektu:
 - w pierwszej linii jak dla klas `Person`
 - w drugiej linii dane matki w formacie "mother: (firstName) (familyName) (age)"
 - w trzeciej linii dane ojca w formacie "father: (firstName) (familyName) (age)". Jesli ojciec nie jest znany "father: no data"

Pozostałe wymagania odczytasz z przypadków testowych. Musisz zapisać kod tak, aby każdy z przypadków się komplował oraz produkował oczekiwane wyniki.

Na przykład:

Test	Wynik
/* Test: poprawne tworzenie obiektu Person, dane poprawne */ try { Person p = new Person("Holenda", "Krzysztof", age: 18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }	Krzysztof Holenda (18)
/* Test: poprawne tworzenie obiektu Person, błędne imię lub nazwisko, piersiowa data pozostała taka sama */ try { Person p = new Person("Olenda", "Krzysztof", age: 18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }	Krzysztof Molenda (18)
/* Test: poprawne tworzenie obiektu Person, błędne imię lub nazwisko, niemallestny wiek */ try { Person p = new Person("Holenda", "Krzysztof", age: -18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }	Age must be positive!
/* Test: modyfikacja obiektu */ try { Person p = new Person("Holenda", "Krzysztof", age: 18); p.modifyFirstName(" Jan "); p.modifyFamilyName(" Kolenda "); p.modifyAge(35); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }	Jan Kolenda (35)
/* Test: tworzenie obiektu Child */ try { Person o = new Person("Holenda", "Krzysztof", age: 30); Person m = new Person("Holenda", "Ewa", age: 29); Child d = new Child("Holenda", "Anna", age: 10, mother: o, father: m); Console.WriteLine(d); } catch(Exception e) { Console.WriteLine(e.Message); }	Anna Holenda (10) mother: Ewa Holenda (29) father: Krzysztof Holenda (30)
/* Test: tworzenie obiektu Child brak jednego z rodziców */ try { Person o = new Person("Holenda", "Krzysztof", age: 30); Person m = new Person("Holenda", "Ewa", age: 29); Child d = new Child("Holenda", "Anna", age: 14, mother: o); d = new Child("Holenda", "Anna", age: 14, mother: m); Console.WriteLine(d); } catch(Exception e) { Console.WriteLine(e.Message); }	Anna Holenda (14) mother: No data father: Krzysztof Holenda (30) Anna Holenda (14) mother: Ewa Holenda (29) father: No data
/* Test: tworzenie obiektu Child brak obu rodziców */ try { Person o = new Person("Holenda", "Krzysztof", age: 30); Person m = new Person("Holenda", "Ewa", age: 29); Child d = new Child("Holenda", "Anna", age: 14, mother: o, father: m); Console.WriteLine(d); } catch(Exception e) { Console.WriteLine(e.Message); }	Anna Holenda (14) mother: No data father: Krzysztof Holenda (30) Anna Holenda (14) mother: Ewa Holenda (29) father: No data

Odpowiedź: (system kar: 0 %)

Zresetuj odpowiedź

```

48     for (int i = 0; i < tekst.Length; i++)
49     {
50         if (tekst[i] != ' ')
51             tekstTemp += tekst[i];
52     }
53     return tekstTemp;
54 }
55
56 private static string WielkosciLitere(string tekst)
57 {
58     string tekstTemp = String.Empty;
59     for (int i = 0; i < tekst.Length; i++)
60     {
61         if (i == 0)
62             tekstTemp += tekst[i].ToString().ToUpper();
63         else
64         {
65             tekstTemp += tekst[i].ToString().ToLower();
66         }
67     }
68     return tekstTemp;
69 }
70
71 private static void SprawdzWiek(int wiek)
72 {
73     if (wiek < 0)
74         throw new ArgumentException("Age must be positive!");
75 }
76
77 public override string ToString()
78 {
79     return $"{FirstName} {FamilyName} ({Age})";

```

Nawigacja w tekście



ZakoncZ przegląd

```

80     }
81 
82     public void modifyFirstName(string imie)
83     {
84         imie = UsunSpacje(imie);
85         SprawdzPoprawnosc(imie);
86         imie = Wielkosciliter(imie);
87 
88         this.firstName = imie;
89     }
90 
91     public void modifyFamilyName(string nazwisko)
92     {
93         nazwisko = UsunSpacje(nazwisko);
94         SprawdzPoprawnosc(nazwisko);
95         nazwisko = Wielkosciliter(nazwisko);
96 
97         this.familyName = nazwisko;
98     }
99 
100    virtual public void modifyAge(int wiek)
101    {
102        SprawdzWiek(wiek);
103        this.Age = wiek;
104    }
105 
106 }
107 
108 class Child : Person
109 {
110     private Person mama { get; }
111     private Person tata { get; }
112     public Child(string firstName, string familyName, int age, Person mother = null, Person father = null) : base(firstName, familyName, age)
113     {
114         if (age > 15)
115             throw new ArgumentException("Child's age must be less than 15!");
116         this.mama = mother;
117         this.tata = father;
118     }
119 
120     public override string ToString()
121     {
122         string wyjście = String.Empty;
123         wyjście += $"{firstName} {familyName} ({Age})";
124 
125         if (mama == null)
126             wyjście += $"\nmother: No data";
127         else
128             wyjście += $"\nmother: {mama}";
129 
130         if (tata == null)
131             wyjście += $"\nfather: No data";
132         else
133     }

```

Test	Oczekiwane	Otrzymane	
/* Test: poprawne tworzenie obiektu Person, dane poprawne */	Krzesztof Holenda (18)	Krzesztof Holenda (18)	✓
try { Person p = new Person(familyName: "Holenda", firstName: "Krzysztof", age: 18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }			
/* Test: poprawne tworzenie obiektu Person, błędne imię lub nazwisko, spacje przed i po */	Krzesztof Holenda (18)	Krzesztof Holenda (18)	✓
try { Person p = new Person(familyName: " Holenda ", firstName: " Krzysztof ", age: 18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }			
/* Test: poprawne tworzenie obiektu Person, błędne imię lub nazwisko, spacje w środku */	Wrong name!	Wrong name!	✓
try { Person p = new Person(familyName: "Molen da", firstName: "Krzysztof.", age: 18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }			
/* Test: poprawne tworzenie obiektu Person, błędne imię lub nazwisko, nie tylko litery */	Wrong name!	Wrong name!	✓
try { Person p = new Person(familyName: "Holenda", firstName: "Krzysztof.", age: 18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }			
/* Test: poprawne tworzenie obiektu Person, błędne imię lub nazwisko, pierswsza duża pozostałe małe */	Krzesztof Holenda (18)	Krzesztof Holenda (18)	✓
try { Person p = new Person(familyName: "Holenda", firstName: "krzysztof", age: 18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }			
/* Test: poprawne tworzenie obiektu Person, błędne imię lub nazwisko, niewiadomy wiek */	Age must be positive!	Age must be positive!	✓
try { Person p = new Person(familyName: "Holenda", firstName: "Krzysztof", age: -18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }			
/* Test: poprawne tworzenie obiektu Person, błędne imię lub nazwisko, pusty napis */	Wrong name!	Wrong name!	✓
try { Person p = new Person(familyName: " 4. ", firstName: "Krzysztof", age: 18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }			
/* Test: modyfikacja obiektu */	Jan Kolenda (35)	Jan Kolenda (35)	✓
try { Person p = new Person(familyName: "Holenda", firstName: "Krzysztof", age: 18); p.modifyFirstName(" Ja "); p.modifyFamilyName("kolenda "); p.modifyAge(35); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }			
/* Test: tworzenie obiektu Child */	Anna Holenda (18) mother: Ewa Holenda (29) father: Krzesztof Holenda (30)	Anna Holenda (18) mother: Ewa Holenda (29) father: Krzesztof Holenda (30)	✓
try { Person o = new Person(familyName: "Holenda", firstName: "Krzysztof", age: 30); Person m = new Person(familyName: "Holenda", firstName: "Ewa", age: 20); Child d = new Child(familyName: "Holenda", firstName: "Anna", age: 10, mother: m, father: o); Console.WriteLine(d); } catch(Exception e) { Console.WriteLine(e.Message); }			
/* Test: tworzenie obiektu Child zły wiek */	Child's age must be less than 15!	Child's age must be less than 15!	✓
try { Person o = new Person(familyName: "Holenda", firstName: "Krzysztof", age: 30); Person m = new Person(familyName: "Holenda", firstName: "Ewa", age: 20); Child d = new Child(familyName: "Holenda", firstName: "Anna", age: 18, mother: m, father: o); Console.WriteLine(d); } catch(Exception e) { Console.WriteLine(e.Message); }			

✓ /* Test: tworzenie obiektu Child złynick */ try { Person o = new Person(familyName: "Holenda", firstName: "Krzysztof", age: 30); Person m = new Person(familyName: "Holenda", firstName: "Ewa", age: 29); Child d = new Child(familyName: "Holenda", firstName: "Anna", age: -15, mother: o, father: o); Console.WriteLine(d); } catch(Exception e) { Console.WriteLine(e.Message); }	Age must be positive!	Age must be positive!	✓
✓ /* Test: tworzenie obiektu Child brak rodziców */ try { Child d = new Child(firstName: "Anna", age: 14); Console.WriteLine(d); } catch(Exception e) { Console.WriteLine(e.Message); }	Anna Holenda (14) mother: No data father: No data	Anna Holenda (14) mother: No data father: No data	✓
✓ /* Test: tworzenie obiektu Child brak jednego z rodziców */ try { Person o = new Person(familyName: "Holenda", firstName: "Krzysztof", age: 30); Person m = new Person(familyName: "Holenda", firstName: "Ewa", age: 29); Child d = new Child(familyName: "Holenda", firstName: "Anna", age: 14, father: o); Console.WriteLine(d); d = new Child(familyName: "Holenda", firstName: "Anna", age: 14, mother: m); Console.WriteLine(d); } catch(Exception e) { Console.WriteLine(e.Message); }	Anna Holenda (14) mother: No data father: Krzysztof Holenda (30) Anna Holenda (14) mother: Ewa Holenda (29) father: No data	Anna Holenda (14) mother: No data father: Krzysztof Holenda (30) Anna Holenda (14) mother: Ewa Holenda (29) father: No data	✓
✓ /* Test: tworzenie obiektu Child modyfikacja danych */ try { Person o = new Person(familyName: "Holenda", firstName: "Krzysztof", age: 30); Person m = new Person(familyName: "Holenda", firstName: "Ewa", age: 29); Child d = new Child(familyName: "Holenda", firstName: "Anna", age: 14, mother: m, father: o); d.modifyFirstName("Aneta"); Console.WriteLine(d); d.modifyFirstName("Holenda"); Console.WriteLine(d); d.modifyAge(18); Console.WriteLine(d); } catch(Exception e) { Console.WriteLine(e.Message); }	Aneta Holenda (14) mother: Ewa Holenda (29) father: Krzysztof Holenda (30) Krzysztof Holenda (18) mother: Ewa Holenda (29) father: Krzysztof Holenda (30) Child's age must be less than 15!	Aneta Holenda (14) mother: Ewa Holenda (29) father: Krzysztof Holenda (30) Krzysztof Holenda (18) mother: Ewa Holenda (29) father: Krzysztof Holenda (30) Child's age must be less than 15!	✓
✓ /* Test: poprawne tworzenie obiektu Person, błędne imię lub nazwisko, specje w środku */ try { Person p = new Person(familyName: "Holen da", firstName: "Krzysztof", age: 18); Console.WriteLine(p); } catch(Exception e) { Console.WriteLine(e.Message); }	Krzysztof Holenda (18)	Krzysztof Holenda (18)	✓

Przeszedł wszystkie testy! ✓

Poprawne
Punkty dla tej odpowiedzi: 5,00/5,00.

Zakończ przegląd

Ćwiczenie: konto w banku (nie oceniane) ►

◀ Zadanie: KM003Z01 - Czas24h

Przejdź do...



Jestes zalogowany(a) jako Petek Krystian [Wyloguj]
wfp-2020-2-1-hb-k-Holenda
Podsumowanie zasad przechowywania danych