

# Technologie .NET

Kolokwium, 05.05.2011

## Grupa A

1. (6pkt) Przeanalizuj poniższy kod, a następnie napisz, co program wypisze na konsolę.

```
class Program
{
    static void Main(string[] args)
    {
        A a1 = new A(1); Console.WriteLine(a1.S + " " + a1.I);
        A a2 = new A() { S = "A2" }; Console.WriteLine(a2.S + " " + a2.I);
        A a3 = a2.New(); Console.WriteLine(a3.S + " " + a3[4]);
    }
}
class A
{
    public String S { get; set; }
    static int i = 5;
    public int I { get { return i += 7; } }
    public int this[int j] { get { return I / j; } }
    public A(int k) { S = "A0"; i -= k; }
    public A() : this(i) { i %= 3; }
    public A New() { return i % 4 > 1 ? this : new A(); }
}
```

---

**A0 11**

**A2 7**

**A2 3**

---

2. (10pkt) Przeanalizuj poniższy kod, a następnie znajdź i krótko opisz 5 błędów, które zostaną wykryte podczas kompilacji. O ile to możliwe, w odpowiedzi wskaż linię, w której wystąpił błąd.

```
00: class A
01: {
02:     protected string s = "A";
03:     public static int N = 2;
04:     public string f1(int i) { return s + i; }
05: }
06: class B : A
07: {
08:     double d;
09:     protected int k;
10:     public new string f1(int i) { return s + i / 2; }
11:     public virtual string f2(string s) { return this.s + s; }
12:     public virtual int f3(int i) { return i++; }
13: }
14: class C : B
15: {
16:     public sealed override string f2(string s) { return s + d; }
17:     public sealed override int f3(int i) { return base.f3(i) + 2; }
18: }
```

```

19: abstract class D : C
20: {
21:     public new string f2(string s) { return this.N; }
22:     public abstract void f4();
23: }
24: class E : D
25: {
26:     public override int f3(int i) { return k - 1; }
27:     public static bool operator ==(E e1, E e2) { return e1.Equals(e2); }
28: }

```

---

**16: zmienna *d* nie jest dziedziczona, ponieważ jest prywatna**

**21: nie można się odwołać do statycznej zmiennej *N* na rzecz obiektu ('this')**

**24: w klasie *E* brak definicji dziedziczonej metody abstrakcyjnej *f4()***

**26: metoda *f3(int)* nie może zostać przesłonięta, ponieważ jest zamknięta w *C***

**27: definiując operator '==' należy również zdefiniować operator '!='**

\* W rzeczywistości w zadaniu znalazł się jeszcze jeden błąd – w linii 21 metoda *f2* powinna zwrócić wartość typu *string*, natomiast zmienna *N* jest typu *int*, a w tym miejscu nie możemy dokonać konwersji niejawniej. W związku z tym punkty będą przyznane za każdy znaleziony błąd – zatem wystarczyło wskazać 5 z 6 powyższych błędów, żeby zdobyć maksymalną liczbę punktów

---

3. (10pkt) Przeanalizuj poniższy kod, a następnie znajdź i krótko opisz 5 błędów, które zostaną wykryte podczas kompilacji. O ile to możliwe, w odpowiedzi wskaż linię, w której wystąpił błąd.

```

00: class Program
01: {
02:     static void Main(string[] args)
03:     {
04:         A a1 = new A();
05:         A a2 = new B();
06:         B b1 = new B();
07:         B b2 = new C();
08:         C c1 = new B();
09:         C c2 = new C() { L = 1 };
10:         C c3 = new C() { k = 2, L = 4 };
11:         b1.f1();
12:         b2.f2();
13:         b2.f3();
14:         c2.L = 7;
15:         c2.K = 3;
16:         c2.f2();
17:         c2.f3();
18:     }
19: }
20: abstract class A
21: {
22:     protected int k;
23:     public int K { get { return k; } }
24:     public int L { get; set; }
25:     public void f1() { Console.WriteLine(K + L); }
26: }
27: class B : A { public void f2() { f1(); } }
28: class C : B { public void f3() { (new C() { k = 3, L = 5 }).f2(); } }

```

---

**04: nie można utworzyć instancji klasy abstrakcyjnej A**

**08: nie można niejawnie rzutować obiektu klasy B na typ C**

**10: zmienna k jest chroniona i nie jest dostępna w tym kontekście**

**13: zmienna b2 jest typu B, który nie zawiera definicji metody f3()**

**15: właściwość K jest tylko do odczytu**

---

4. (15pkt) Przeanalizuj poniższy kod, a następnie napisz, co program wypisze na konsolę.

```
class Program
{
    static void Main(string[] args)
    {
        A.K = 1;
        A a = new A(); B b = new B(); C c = new C(); D d = new D();
        Console.WriteLine(a.f() + " " + b.f() + " " + c.f() + " " + d.f());
        A.K = 2;
        a = new B(); b = new C(); c = new D();
        Console.WriteLine(a.f() + " " + b.g() + " " + c.h());
        A.K = 3;
        a = new C(); b = new D();
        Console.WriteLine(((C)a).h() + " " + ((D)b).h());
        A.K = 4;
        a = new D();
        Console.WriteLine(((C)a).f());
    }
}
class A
{
    public static int K;
    public virtual int f() { return K -= 2; }
}
class B : A
{
    public override int f() { return K += 3; }
    public virtual int g() { return f() - 1; }
}
class C : B
{
    public new virtual int f() { return K += 2; }
    public virtual int h() { return f() + g(); }
}
class D : C
{
    public override int f() { return h() - 2; }
    public override int h() { return base.f() - g(); }
}
```

---

**-1 2 4 -4**

**5 7 -2**

**12 -2**

**-4**

---

5. (9pkt) Przeanalizuj poniższy kod, a następnie napisz, co program wypisze na konsolę.

```
class Program
{
    static void Main(string[] args)
    {
        A a = new A();
        a.Get += f;
        a.Set += g;
        a.K = 5;
        Console.WriteLine(a.K);
        a.Get += g;
        a.Set += A.I;
        a.K = 10;
        Console.WriteLine(a.K);
        a.Get -= f;
        a.Set -= A.I;
        a.K = 7;
        Console.WriteLine(a.K);
    }
    static int f(int n) { return n + 2; }
    static int g(int n) { return n - 1; }
    static int h(int n) { return n * 2; }
}
class A
{
    private int k = 0;
    public int K { get { return Get(k); } set { k = Set(value); } }
    public delegate int F(int n);
    public event F Get;
    public event F Set;
    public static int I(int n) { return n % 5; }
}
```

---

6

-1

5

---