1. Co to jest zmienna
    1. symbol określonego typu, pod którym kryje się pewna wartość
    2. wartość (liczbowa, tekstowa, itp.), która jest znana na etapie kompilacji i nie może się nigdy zmienić
    3. wydzielona część programu wykonująca jakieś operacje
2. Definiując zmienną należy koniecznie:
    1. określić jej typ i nazwę
    2. określić jej nazwę i wartość
    3. określić jej typ, nazwę i wartość.
3. Do czego służy funkcja Main
    1. do niczego szczególnego, można ją pominąć w programie
    2. jest to miejsce programu, od którego zaczyna on swoje działanie
    3. jest to funkcja, która odpowiada za przygotowanie kompilatora do pracy
4. Co to jest algorytm?
    1. Skompilowany kod programu
    2. przepis na rozwiązanie problemu
    3. problem, do którego należy znaleźć rozwiązanie
5. Jaki będzie wynik poniższego kodu (petla)

```
int ile = 0;
for (int i = 1; i <= 10; ++i)
{
if (i % 2 == 0)
{
ile++;
suma += i;
}
}
suma /= ile;
Console.WriteLine(suma);
```

    1. 30
    2. 6
    3. 5
    4. 5.5
6. Czym jest tablica?
    1. jest to ciąg zmiennych jednego typu. Posiada jedną nazwę a do jego poszczególnych elementów odnosimy się przez numer (indeks).
    2. jest to ciąg zmiennych różnych typów. Posiada jedną nazwę a do jego poszczególnych elementów odnosimy się przez numer (indeks).
    3. jest to ciąg zmiennych jednego typu. Posiada jedną nazwę i nie możemy odwołać się bezpośrednio do pojedynczego elementu.
7. Który kod poprawnie wypełni tablicę losowymi liczbami z przedziału <1; 10>

```
ar = new int[8];
Random rnd = new Random();
1. for (int i = 0; i < ar.Length; ++i) ar[i] = i + 1;
2. for (int i = 0; i < ar.Length; ++i)
        ar[i] = int.Parse(Console.ReadLine());
3. for (int i = 0; i < ar.Length; ++i) rnd.Next(1, 11);
4. for (int i = 0; i < ar.Length; ++i) rnd.Next(1, 10);
```

8. Czym nie jest klasa?
    1. nowym typem danych
    2. fragmentem pamięci, któremu przyporządkowano pewien typ
    3. grupą danych i funkcji, które na tych danych operują
9. Jaka jest różnica między klasą a obiektem?
    1. Nie ma żadnej.
    2. Klasa definiuje nowy typ danych, a obiekt oznacza zmienną zgodną z opisem klasy
    3. Obiekt definiuje nowy typ danych, a klasa oznacza zmienną zgodną z opisem obiektu
10. Jaka jest różnica między interfejsem a implementacją klasy?
    1. Nie ma żadnej
    2. Interfejs klasy stanowi funkcjonalność, z jakiej można skorzystać a implementacja jest prywatną sprawą twórcy klasy i sprawia, że interfejs działa.

3. Twórca klasy nie powinien zmieniać implementacji klasy bez ostrzeżenia, zaś co zrobi z interfejsem jest jego prywatną sprawą

11. Co to jest konstruktor?
    1. Jest to jedna ze składowych klasy
    2. jest to metoda w klasie, dzieki której możliwe jest utworzenie obiektu
    3. jest to w klasie o takiej samej nazwie jak klasa i zwracająca wartość całkowitą
12. Co należałoby poprawić w poniższym kodzie, aby zadziałał?

```
class Punkt {
   private float x, y;
   public override string ToString()
   {
      return "(" + x + ", " + y + ")";
   }
   public void Przesuń(int dx, int dy)
   {
      x += dx; y += dy;
   }
}

class Program
{
      static void Main(string[] args)
      {
            Punkt p1 = new Punkt(3, 2);
            p1.Przesuń(1, 0);
            Console.WriteLine(p1);
      }
}
```

    1. zmienić kod metody ToString
    2. dopisać konstruktor parametryczny
    3. metodę Przesun zrobic prywatną
13. Dana jest klasa:

```
class Osoba{
   private string imie, nazwisko;
   private double pensja;
   ...
}
```

    Która z poniższych implementacji metody ToString pozwoli na poprawne wypisanie wypisanie tekstu typu: "Jan Kowalski zarabia 2500 PLN"?

```
1. public override void ToString() {
   Console.WriteLine(imie+" "+nazwisko+" zarabia "+ pensja+" PLN.");
   }
2. private override string ToString() {
   return imie+" " + nazwisko + " zarabia " + pensja+" PLN.";
   }
3. public override string ToString() {
   return imie+" " + nazwisko + " zarabia " + pensja+" PLN.";
   }
4. public override string ToString() {
   string s = imie+" " + nazwisko + " zarabia " + pensja+" PLN.";
   }
```

14. Jaka jest różnica między metodą a funkcją?
    1. metoda nie może mieć parametrów wejściowych, a funkcja może
    2. metoda nie może mieć zmiennych lokalnych, a funkcja może
    3. nie ma żadnej różnicy
    4. metoda jest wykonywana przez konkretny obiekt, a funkcja nie ma jasno określonego wykonawcy
15. Co to jest konstruktor domyślny?
    1. pierwszy napisany konstruktor
    2. konstruktor o największej wadze
    3. konstruktor dodany automatycznie do klasy, jeśli nie ma ona żadnego konstruktora
    4. dla danego obiektu – konstruktor użyty do jego stworzenia
16. Ile konstruktorów może mieć klasa?

1. wiele o tej samej nazwie, pod warunkiem, że mają różne listy parametrów
2. tylko jeden
3. wiele, pod warunkiem, że mają różne nazwy

17. Co to jest interfejs klasy?
    1. zbiór nagłówków oferowanych przez klasę publicznych metod, które można z niej wywołać
    2. pełna nazwa klasy
    3. pełny kod klasy
18. Gdzie dostępne są składowe prywatne danej klasy?
    1. wszędzie w programie
    2. jedynie w konstruktorach
    3. wewnątrz tej klasy
    4. jedynie w metodach klasy
19. Do czego służy mechanizm hermetyzacji?
    1. w zasadzie do niczego, jest to wymysł niektórych zwariowanych programistów
    2. do jasnego wskazania, co jest interfejsem, a co implementacją klasy
    3. do ukrycia w pamięci tajnych informacji, których ujawnienie naraziłoby użytkownika programu na duże straty (hasło dostępu do komputera, nr karty kredytowej)
    4. do przyspieszenia działania programów
20. Co to jest implementacja klasy?
    1. wszystko to, co sprawia, że interfejs działa
    2. archiwum ze skompilowaną wersją i kodem źródłowym klasy
    3. treść (kod) oferowanych metod z interfejsu , pola klasy oraz inne, pomocnicze metody, wywoływane przez metody z interfejsu
    4. cały kod klasy (z wyjątkiem jej interfejsu)
21. Do czego służy technika kompozycji?
    1. do łatwiejszego tworzenia nowych klas z użyciem klas istniejących
    2. do tworzenia klas bibliotecznych
    3. do łatwiejszego tworzenia nowych obiektów
22. Do czego służy metoda ToString?
    1. do wypisania na ekranie zawartości wszystkich pól obiektu
    2. do zwrócenia tekstowego opisu obiektu
    3. do automatycznego wypisania kodów wszystkich metod obiektu
23. Która relacja między klasą bazową z pochodną nie jest prawdziwa
    1. klasa pochodna jest szczególnym przypadkiem klasy bazowej
    2. klasa pochodna jest rodzajem klasy bazowej
    3. klasa bazowa jest uogólnieniem klasy pochodnej
    4. klas pochodna zawiera klasę bazową
24. Na co wskazuje pseudozmienna 'base' w C#?
    1. na pierwsze pole obiektu
    2. na obiekt klasy bazowej zawarty w obiekcie klasy pochodnej
    3. na konstruktor klasy bazowej
25. Czy składowe prywatne są dziedziczone?
    1. prywatne metody są dziedziczone, a prywatne pola nie
    2. tak i klasa dziedzicząca może z nich korzystać bez ograniczeń
    3. tak, ale klasa dziedzicząca nie może z nich bezpośrednio korzystać
    4. nie
    5. prywatne pola są dziedziczone, a prywatne metody nie
26. Co to jest klasa bazowa?
    1. klasa, z której dziedziczy inna klasa
    2. klasa, która dziedziczy z innej klasy
    3. klasa, która zawiera definicję innej klasy
    4. klasa z metodą główną (main)
27. Co to jest klasa pochodna?
    1. klasa z metodą główną (main)
    2. klasa, która zawiera definicję innej klasy
    3. klasa, z której dziedziczy inna klasa

4. klasa, która dziedziczy z innej klasy

28. Czy Kwadrat powinien dziedziczyć z Prostokąta, czy Prostokąt z Kwadratu?
    1. Prostokąt powinien dziedziczyć z Kwadratu, bo każdy Prostokąt jest też Kwadratem
    2. Kwadrat powinien dziedziczyć z Prostokąta, bo każdy Kwadrat jest też Prostokątem
    3. Prostokąt powinien dziedziczyć z Kwadratu, bo Kwadrat ma jeden bok 'a', a Prostokąt dwa: 'a' i 'b'
    4. może być i tak, i tak
29. Co się dzieje, kiedy klasa definiuje własne pole o tej samej nazwie co pole odziedziczone?
    1. nie ma z tym problemu – pola o tej samej nazwie to pola przeciążone
    2. nie dziedziczy z klasy bazowej pól o takich samych nazwach, co pola na nowo zdefiniowane w klasie pochodnej
    3. nie jest to możliwe – wystąpi błąd kompilacji
    4. obiekt klasy pochodnej ma kilka pól o tej samej nazwie, pola własne są „ważniejsze” niż pola odziedziczone
30. Co to jest przesłanianie składowych w dziedziczeniu?
    1. to zmiana nazwy dziedziczonych składowych
    2. to usuwanie wybranych składowych odziedziczonych
    3. to dziedziczenie tylko wybranych składowych z klasy bazowej
    4. to definiowanie w klasie dziedziczącej składowych o tej samej nazwie co składowej odziedziczone
31. Czego klasa nie dziedziczy?
    1. konstruktorów
    2. metod
    3. pól
32. Czy klasa ma dostęp do przesłoniętych pól?
    1. nie, definiowanie pól o tej samej nazwie co pola odziedziczone powoduje błąd kompilacji
    2. tak, do wszystkich swoich pól ma bezpośredni dostęp
    3. tak, z wyjątkiem pól prywatnych
    4. tak, ale tylko do pól publicznych