

1. Który z poniższych języków programowania jest językiem programowania obiektowego?

- a) ML
- b) C#**
- c) Haskell
- d) Prolog

2. Z rachunku lambda zapis: (M N) oznacza:

- a) zmienną
- b) definicje
- c) aplikacje**
- d) nie ma takiego zapisu w rachunku lambda

3. Która z poniższych komend definiuje zmienną typu prostego w języku C++:

- a) Obiekt *obj;**
- b) int tab[20];
- c) struct Struktura str;
- d) union Unia str;

4. C++ jest językiem programowania w którym:

- a) nie ma systemów typów
- b) system typów jest dynamiczny
- c) system typów jest statyczny**
- d) system typów zawiera tylko typy złożonego

5. Środowisko odniesienia jest to:

- a) zbiór wartości przyjmowanych przez zmienne w programie
- b) zbiór zmiennych w programie
- c) zbiór funkcji zawartych w module programu
- d) zbiór wiązań dostępnych w danym miejscu programu**

6. W typach będących typami rekurencyjnymi:

- a) wartości grupowane są w n-tki
- b) każdej wartości jednego typu odpowiada wartość innego typu
- c) typ ten zawiera sam siebie**
- d) wartość jest wybierana dla jednego z wielu zbiorów

7. Rekord aktywacji funkcji przechowywany jest na:

- a) w odpowiednim pliku systemu operacyjnego
- b) stercie
- c) kolejce
- d) stosie**

8. Dowolna zmienna przechowywana na stercie jest:

- a) zawsze typu prostego
- b) zawsze anonimowa**
- c) zawsze typu złożonego
- d) tworzona zawsze na początku programu

9. Blok programu jest to konstrukcja:

- a) równoważna definicji klasy
- b) ograniczająca zakres widoczności deklaracji wewnętrzniego**
- c) która nie istnieje w nowoczesnych językach programowania
- d) która może łączyć się tylko z instrukcją warunkową

10. Deklaracje równolegle występują w języku:

- a) C#
- b) C++
- c) Scheme**
- d) Prolog

11. W wywołaniu: int s=suma(x,&y) x i y oznaczają:

- a) x parametr formalny, &y parametr aktualny
- b) x parametr aktualny, &y parametr formalny
- c) argumenty funkcji
- d) parametry aktualne funkcji**

12. W języku C++ parametry można przekazać przez wartość:

- a) do funkcji
- b) z funkcji
- c) do i z funkcji**
- d) do lub z funkcji ale nie jednocześnie do i z funkcji

13. W rachunku lambda operacja warunkowa zdefiniowana jest w postaci:

- a) $\lambda mnfx.mf(nfx)$
- b) $\lambda mnf.mnf$**
- c) $\lambda mnfx.mnfx$
- d) $\lambda mnfx.mn(fx)$

14. Kluczową koncepcją paradygmatu programowania logicznego jest:

- a) parametryczny polimorfizm
- b) wywołanie
- c) procedura
- d) relacja**

15. W językach o dynamicznym zakresie widoczności nazw

- a) Funkcja jest wywoływana w kontekście środowiska w którym została zdefiniowana
- b) Funkcja jest wywoływana w kontekście środowiska w którym została wywołana**
- c) Funkcji nie ma
- d) Funkcja jest wykonywana w dowolnym środowisku

16. Domknięcie jest to:

- a) funkcja połączona z leksykalnym środowiskiem tej funkcji**
- b) inna nazwa funkcji pierwszej klasy
- c) konstrukcja programowa występująca w C++
- d) specjalna dyrektywa kompilatora

17. Zredukuj wyrażenie:

$$(\lambda mnfx.mf(nfx) \lambda fx.fx \lambda zy.zzy)$$

$$+ \quad \quad \quad 1 \quad \quad \quad 2$$

$$\underline{\lambda m.(\lambda n.(\lambda fx.mf(nfx))1)2} \quad m \rightarrow 2$$

$$\underline{\lambda n.(\lambda fx.2f(nfx))1} \quad n \rightarrow 1$$

$$\underline{\lambda fx.2f(1fx)} \quad n \rightarrow 1$$

$$\underline{\lambda fx.f(f(f x))} \rightarrow 3$$

1. W pętli

```
for(char i=0; i<256; i++) cout<<i<<endl;
```

instrukcja cout wykona się

a) 256 razy

b) 257 razy

c) więcej niż 257 razy (char => 255 => 100000000 przepelenie!)

d) 255 razy

2. Funkcja rozszerzająca, jest to funkcja

a) pozwalająca rozszerzyć możliwości już istniejącej metody

b) funkcja rozszerzająca kolekcję o nowe elementy

c) funkcja pozwalająca na dostęp do zmiennych prywatnych klasy

d) statyczna

3. Wartości grupowane są w n-tki w:

a) produktach kartezjańskich

b) uniach

c) typach rekurencyjnych

d) mapowaniach

4. Które z poniższych wyrażeń poprawnie definiuje w języku C# wyrażenie lambda

$\lambda x.\lambda y.(x\ y)$

a) $(x,y) \Rightarrow y(x)$

b) $(x,y) \Rightarrow (x\ y)$

c) $(x,y) \Rightarrow (y\ x)$

d) $(x,y) \Rightarrow x(y)$

5. Blok programu jest to konstrukcja

a) Która może się łączyć tylko z instrukcją warunkową

b) Która nie istnieje w nowoczesnych językach programowania

c) Ograniczająca zakres widoczności deklaracji wewnątrz niego

d) Równoważna definicji klasy

6. Która, z poniższych instrukcji deklaruje zmienne sekwencyjne

a) (let* ((x 2) (a 3)) (+ x a))

b) (letrec ((x 2) (a 3)) (+ x a)) (letrec → rekurencyjny)

c) (let ((x 2) (a x)) (+ x a))

d) (let ((x 2) (a 3)) (+ x a))

7. W językach o dynamicznym zakresie widoczności nazw

a) Funkcja jest wywoływana w kontekście środowiska w którym została zdefiniowana

b) Funkcja jest wywoływana w kontekście środowiska w którym została wywołana

c) Funkcji nie ma

d) Funkcja jest wykonywana w dowolnym środowisku

8. **Deklaracje** są to instrukcje, które

a) tworzą nowe funkcje

b) tworzą nowe zmienne

c) tworzą nowe stałe

d) tworzą nowe wiązania

9. Instrukcja typedef języka C++ powoduje

a) przypisanie nowej nazwy do istniejącego typu

b) stworzenie nowej tablicy

c) przypisanie nowej nazwy do istniejącej zmiennej

d) stworzenie nowego typu danych

10. Wyrażenie języka Scheme: '((- 1 2) (- 2 3) (-3 4))

a) Listę wartości: 3, -1, -1, ()

b) Listę funkcji

c) Nie wykona się z powodu błędu składniowego

d) Listę list symboli +, 1, 2, 0; -, 2, 3, 0; -, 3, 4, 0

11. C++ jest językiem programowania, w którym **system typów**:

- a) jest statyczny**
- b) jest dynamiczny
- c) zawiera tylko typy złożone
- d) nie ma go

12. Które wyrażenie rachunku lambda (wzbogaconego instrukcją iloczynu) będzie miało to samo znaczenie, co instrukcja języka

Scheme: (let ((x 2) (y 3)) (* x y))

- a) $\lambda x. \lambda y. (* x y) 3 2$
- b) $\lambda x. [\lambda y. (* x y)] 3 2$
- c) $\lambda x. [\lambda y. (* x y)] 2 3$ jeśli (let* ((x 2) (y 3)) (* x y))
- d) $\lambda x. \lambda y. (* x y) 2 3$**

13. Wyrażenie **((lambda (z) (lambda (x y) (+ x y z))) 1)** zwraca:

- a) Funkcję** (procedura w Scheme)
- b) Domknięcie
- c) Nie wykona się z powodu błędu
- d) Wartość liczbową

14. Zakresem widoczności zmiennych w wyrażeniu **(let ((x 2) (y 3)) (+ x y))** jest:

- a) lista ((x 2) (y 3))
- b) całe wyrażenie (let ((x 2) (y 3)) (+ x y))
- c) wyrażenie (y 3) i (+ x y)
- d) wyrażenie (+ x y)**

15. **Smalltalk** jest językiem **należącym do paradygmatu**:

- a) logicznego
- b) obiektowego**
- c) imperatywnego
- d) funkcyjnego

16. Wyrażenie języka **C#: x=>x+1** tworzy:

- a) wartość logiczną
- b) wartość całkowitą
- c) nie ma takiego wyrażenie w języku C#
- d) Funkcję anonimową (lambda)**

17. Językiem **o dynamicznym systemie typów** nie jest:

- a) Scheme
- b) C**
- c) JavaScript
- d) PHP

18. Do **zwrócenie pierwszego elementu pary w języku Scheme** służy funkcja:

- a) cdr
- b) begin
- c) end
- d) car**

19. Który program języka **PROLOG** pozwoli poprawnie **obliczyć wartość silni**:

- a) silnia(N,W) :- X is N-1, silnia(X, W_N_1), W is N*W_N_1. silnia(0,W) :- W is 1.
- b) silnia(0,W) :- W = 1. silnia(N,W) :- X = N-1, silnia(X, W_N_1), W = N*W_N_1.
- c) silnia(N,W) :- X = N-1, silnia(X, W_N_1), W = N*W_N_1. silnia(0,W) :- W = 1.
- d) silnia(0,W) :- W is 1.**

silnia(N,W) :- X is N-1, silnia(X, W_N_1), W is N*W_N_1.

(, ← koniunkcja (AND) :- ← implikacja ; ← alternatywa (OR))

20. Komenda **mezczyzna(jan)?** języka **PROLOG** definiuje

- a) nowe zapytanie

- b) nowy fakt

- c) ma błąd składniowy**

- d) nową relację

(Zapytanie w Prologu: ?- mezczyzna(jan). ← „Czy Jan jest mężczyzną?”)

21. Wyrażenie **let* języka Scheme** definiuje zmienne

- a) równolegle

- b) rekurencyjnie

- c) których wartość można modyfikować

- d) sekwencyjnie**

22. Wyrażenie **warunkowe w rachunku lambda** zapisuje się jako:

- a) $\lambda stx.tx$

- b) $\lambda stx.stx$**

- c) $\lambda stx.sxt$

- d) $\lambda stx.sx$

true	$\equiv \lambda xy.x$
false	$\equiv \lambda xy.y$
if	$\equiv \lambda bxy.bxy$
not	$\equiv \lambda b.\text{if } b \text{ false true}$
and	$\equiv \lambda b_1b_2.\text{if } b_1 \text{ } b_2 \text{ false}$
or	$\equiv \lambda b_1b_2.\text{if } b_1 \text{ true } b_2$

23. Wyrażenie **let języka Scheme** definiuje zmienne

- a) równolegle**

- b) rekurencyjnie

- c) których wartość można modyfikować

- d) sekwencyjnie

24. Przyjmując, że typ int zajmuje w pamięci 4 bajty, a typ char 1 bajt to zmienna typu union Przykład { int x; char y; }; będzie zajmować:

- a) 4 bajty**

- b) 8 bajtów (struktura int i char)

- c) 6 bajtów

- d) 5 bajtów

25. Wyrażenie języka **C#: x=>x+2** nazywane jest:

- a) funkcją rozszerzającą

- b) funkcją statyczną

- c) nie ma takiego wyrażenia w języku C#

- d) wyrażeniem lambda**

26. **Fakty w PROLOGU:**

- a) są zawsze fałszywe
- b) są albo prawdziwe albo fałszywe
- c) nie mają określonej wartości prawdy

d) są zawsze prawdziwe

27. Słówko kluczowe **var w C#** pozwala na

- a) zdefiniowanie dowolnej zmiennej
- b) zdefiniowanie zmiennej dowolnego typu, który nie może być później zmieniany**
- c) zdefiniowanie zmiennej typu podstawowego Object
- d) zdefiniowanie zmiennej dowolnego typu, który może być później dowolnie zmieniany

28. Zakresem widoczności zmiennych w wyrażeniu

(let* ((x 2) (y 3)) (+ x y)) jest

- a) lista ((x 2) (y 3))
- b) całe wyrażenie (let ((x 2) (y 3)) (+ x y))
- c) wyrażenie (y 3) i (+ x y)**
- d) wyrażenie (+ x y)

29. Który z poniższych języków programowania jest językiem programowania obiektowego:

- a) Haskell
- b) C#**
- c) Prolog
- d) ML

30. **Postać normalna** jest to

a) Wyrażenie, którego nie można dalej redukować

- b) Wyrażenie, które nie zawiera żadnej lambda abstrakcji
- c) Wyrażenie, które nie zawiera zmiennych związanych
- d) Wyrażenie, które nie zawiera zmiennych wolnych

31. **Alfa konwersja** polega na zmianie

a) nazw zmiennych związanych

- b) lambda abstrakcji na aplikację
- c) aplikacji na lambda abstrakcję
- d) nazw zmiennych wolnych

32. **Operator koniunkcji jest w języku PROLOG** oznaczany symbolem:

- a),**
- b);
- c)?
- d).

33. Jaka funkcja zostanie automatycznie stworzona w chwili deklaracji struktury (define-strukt punkt (x y)):

a) make-punkt (konstruktor)

- b) get-punkt (pobiera wartość)
- c) put-punkt
- d) set-punkt (ustawia wartość)

34. Która z poniższych instrukcji poprawnie deklaruje funkcję rozszerzającą:

- a) public static string Funkcja(string str) {...}
- b) public string Funkcja(this string str) {...}
- c) static string Funkcja(this string str) {...}

d) public static string Funkcja(this string str) {...}

(Funkcja rozszerzająca ma this w argumencie – to co rozszerza!)

35. Która z poniższych instrukcji jest **zawsze poprawna**:

- a) (let* ((foo (lambda (x) (bar x))) (bar (lambda (x) (+ 1 x)))) (foo 5))
- b) (letrec ((foo (lambda (x) (bar x))) (bar (lambda (x) (+ 1 x)))) (foo 5))**
- c) (let~ ((foo (lambda (x) (bar x))) (bar (lambda (x) (+ 1 x)))) (foo 5))
- d) (let ((foo (lambda (x) (bar x))) (bar (lambda (x) (+ 1 x)))) (foo 5))

36. Wynikiem operacji **(nfx.f(n f x) st.ssst)** będzie

- a) λst.ssst
 - b) λst.sst
 - c) λfx.ffffx**
 - d) λfx.ffffx
- $\lambda n.(\lambda fx.f(nfx))3$
 $\lambda fx.f(3fx)$
 $\lambda fx.ffffx$

37. W **rekordzie aktywacji funkcji przechowywane są**:

a) zmienne lokalne

- b) stałe
- c) żadne z powyższych
- d) zmienne globalne

38. Do zwrócenie **drugiego elementu pary w języku Scheme** służy funkcja

a) cdr

- b) begin
- c) end
- d) car

39. Przy przekazywaniu przez wartość argumentów "z" funkcji parametrem aktualnym może być:

- a) stała
- b) zmienna (przez &)
- c) dowolnie**

40. Zakresem widoczności zmiennych w wyrażeniu (**letrec ((x 2) (y 3)) (+ x y)**) jest
a) lista ((x 2) (y 3))

b) całe wyrażenie (let ((x 2) (y 3)) (+ x y))

c) wyrażenie (y 3) i (+ x y)

d) wyrażenie (+ x y)

41. Dowolna **zmienna przechowywana na stercie** jest (na stercie są wskaźniki!)

a) zawsze typu złożonego

b) zawsze anonimowa

c) tworzona zawsze na początku programu

d) zawsze typu prostego

42. Wyrażenie **letrec języka Scheme** definiuje zmienne

a) równolegle

b) rekurencyjnie

c) których wartość można modyfikować

d) sekwencyjnie

43. **Zmienna związana** jest to

a) zmienna do której jest przypisana wartość

b) zmienna, która występuje w ciele funkcji w której jest zdefiniowana

c) która nie istnieje, gdyż istnieją tylko zmienne wolne

d) zmienna do której przypisana jest inna zmienna

44. Program

```
#include <iostream>
using namespace std;
template <typename T>
T max(T x, T y)
{ return (x > y)? x : y;}
int main()
{
    cout << max(3, 7) << std::endl;
    cout << max(3.0, 7.0) << std::endl;
    cout << max(3, 7.0) << std::endl;
    return 0;
}
```

da w rezultacie

a) Błąd komplikacji ze względu na brak typu danych w wywołaniu funkcji szablonowej

b) Błąd komplikacji ze względu na niejednoznaczność trzeciego wywołania funkcji max

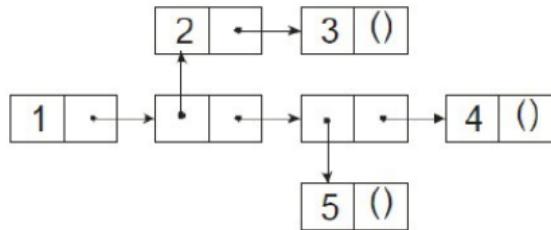
c) Żadne z powyższych (przysłonięcie funkcji max, bo jest std)

d) wartości 7, 7.0, 7.0

45. Która, z poniższych instrukcji występuje tylko w paradygmacie imperatywnym i pochodnych

- a) wywołania funkcji
- b) deklaracji
- c) warunkowa
- d) pętli**

46. Za pomocą którego wyrażenia języka Scheme uzyskano poniższą strukturę



- a) (cons 1 (cons (cons 2 (cons 3 empty)) (cons (cons 5 empty) (cons 4 empty))))**
- b) (list 1 2 3 4 5)
- c) (cons 1 (cons (cons 2 3) (cons 5 4))))
- d) (cons 1 (cons 2 (cons 3 (cons 4 (cons 5 empty))))))

Operator » w F# służy do:

Tworzenia nowej funkcji poprzez złączenie dwóch istniejących

W F# iloczyn kartezjański, w którym składowe mają swoją nazwę nazywamy:
rekordem

Która deklaracja tworzy funkcję sumującą elementy na liście:

```
let rec sumuj = function | [] → 0 | x::xs → x + sumuj xs
```

W celu zastosowania walidacji rekordu F#, która zwraca wszystkie komunikaty o błędach jednocześnie, najlepiej zastosować wzorzec
monady

N-tką (krotką) w F# nazywamy uporządkowany zbiór wartości w którym:
elementy są tego samego lub różnych typów

Niech dane będą funkcje:

```
let liczbaZnaków str = Some(String.length str)
```

```
let razy2 x = Some(x*2)
```

ich złożenia można dokonać za pomocą:

żadne z wymienionych

Rekordy w C# mogą dziedziczyć:

Tylko po rekordach

Brak wartości może być obsłużony w F# za pomocą typu:

Option

W programowaniu czysto obiektowym funkcje wyższych rzędów można zastąpić zastosowaniem wzorca projektowego:

Strategia

Typ danych reprezentujący komendę w C# nazywa się:

Action

Z pomocą której instrukcji możemy zmienić składową imię w rekordzie zdefiniowanym jako: type Osoba = {imię:string,nazwisko:string}

żadne z wymienionych

Dana jest funkcja let f x y = x+y co zostanie zwrócone w przypadku następującego jej wywołania:

f 1

inna funkcja

W celu obsługi błędów w programach funkcyjnych można użyć

Monady Maybe lub Option

W Linq operacja mapowania realizowana jest za pomocą metody:

Select

Ilu elementową tablicę zwróci wyrażenie C#

```
Enumerable.Range(0,10).ToArray().Select(n=>n*n);
```

Nie zwróci tablicy

Jeżeli chcesz aby w interface iMoj<T>{} zmienna typu T mogła być wykorzystana tylko jako rezultat metody lub właściwości opiszesz ją jako:

out

Ile argumentów należy przekazać do poniższej funkcji let f x = function Some x → x+y | None → x, aby się całkowicie wykonała

2

W języku C# funkcja generująca wartość opisaną typem Lazy<T>:

powinna być czysta/powinna zawsze posiadać efekty uboczne

Który z poniższych deklaracji wymusi na programiście konieczność podania wszystkich argumentów funkcji jednocześnie:

let dodaj (x,y,z) = x+y+z

Który z poniższych zapisów przedstawia sygnaturę funkcji złącz (bind) dla typu Option<'T>:

złącz: Option<'T> → ('T → Option<'R>) → Option<'R>