6. Ниже приведена спецификация родового пакета Stacks на языке Ада.

Напишите объявление шаблонного класса на языке Си++, предназначенного

для той же роли, что и этот пакет.

GENERIC

TYPE T IS PRIVATE; SIZE : INTEGER;

PACKAGE Stacks IS

TYPE Stack IS LIMITED PRIVATE;

PROCEDURE Push(S: IN OUT Stack; X : IN T);

PROCEDURE Pop(S: IN OUT Stack; X : OUT T);

FUNCTION IsEmpty(S : IN Stack) RETURN BOOLEAN;

FUNCTION IsFull(S : IN Stack) RETURN BOOLEAN;

PRIVATE

TYPE Stack is RECORD

Body : ARRAY (1..SIZE) OF T;

Top : INTEGER := 1;

END RECORD;

END Stacks;

Один из вариантов:

template <typename T, int size> class Stack

{

public:

Stack() {top = 0;}

void Push(T x);

T Pop(T& x);

bool IsEmpty();

bool IsFull();

private:

Stack (const Stack& s);

T body[N];

int top;

};

8. Напишите на языке Ада 95 объявления, эквивалентные

приведенным ниже описаниям на языке Оберон-2.

TYPE T\* = RECORD I\*, J : INTEGER; END;

TYPE T1\* = RECORD (T) K : INTEGER; END;

PROCEDURE (VAR X: T) P\* (L : INTEGER);

PROCEDURE (VAR X: T1) P\* (L : INTEGER);

Полностью эквивалентный фрагмент написать нельзя, поскольку Ада

требует полной инкапсуляции структуры типа, а Оберон позволяет открывать

поля структуры (в примере - I открыто, а J – закрыто). Однако можно на Аде

написать написать подпрограммы доступа для I (get/set) и добиться того же

эффекта.

type T is tagged private;

type T1 is new tagged T with private;

procedure P(X:T; L: integer);

procedure P(X:T1; L: integer);

Замечание: на языке Оберон процедуры P динамически привязаны к

типу (T и T1 соответственно), однако на Аде динамическая привязка — это

свойство не метода, а вызова. Поэтому разницы между динамически и

статическими привязанными методами в Аде нет.

3. Напишите спецификацию абстрактного типа данных Deque (очередь

с двумя “хвостами”) на языках Ада и Java (тела методов и тело пакета можно

опустить).

Язык Ада (реализация в виде двунаправленного списка):

generic

type T is private;

package G\_Deque is

type Deque is limited private;

procedure PushRight(Deq: inout Deque; X:T);

procedure PushLeft(Deq: inout Deque; X:T);

procedure PopRight(Deq: inout Deque; X: out T);

procedure PopLeft(Deq: inout Deque; X: out T);

procedure Init(Deq: out Deque);

procedure Destroy(Deq: inout Deque);

function IsFull(Deq: Deque);

function IsEmpty(Deq: Deque);

–- другие процедуры ...

private

type PLink is access;

type Link is record inf : T; next, prev : PLink; end record;

type PLink is access Link;

type Deque is record Left, Right: PLink; end record;

end G\_Deque;

Язык Java;

interface IDeque<T>

{

void PushLeft(T x);

void PushRight(T x);

T PopLeft();

T PopRight();

bool IsFull();

bool IsEmpty();

// другие функции

}

Замечания:

а). Обобщенные конструкции употреблять не обязательно (надо только

написать, что тип Т должен быть непосредственно видимым в точке

описания типа Deque). Хотя обобщения здесь подходят больше.

б). Структуру типа в Аде полностью выписывать необязательно.

Главное — указать наличие приватной части, например:

private

...

type Deque is …; –- структура типа Deque

end G\_Deque;

в). Для языка Java можно выписать не инетерфейс, а конкретный класс

с приватной структурой и публичными функциями-операциями. Тела

функций в этом случае можно не выписывать. Но интерфейс в данном случае

больше подходит к понятию абстрактного типа данных.

7. Ниже приведена спецификация шаблонной функции перемножения

матриц(двумерных массивов) на языке Си++. Напишите пример

конкретизации этой функции, а также соответствующее описание родовой

функции на языке Ада.

template <class T> Matrix<T>& MatMult (Matrix<T>& A,

Matrix<T>&B);

Конкретизация на языке Си++:

Matrix<float> b,c;

…

Matrix<float> a = MatMult(b,c);

Язык Ада:

generic

type T is private;

with function “+”(x,y:T) return T (<>);

with function “\*”(x,y:T) return T (<>);

type Matrix is private;

function G\_MatMult(A,B: Matrix) return Matrix;

7. Опишите на языке Ада родовой модуль, реализующий абстрактный

тип данных Queue (очередь). Реализацию процедур и функций писать не

надо.

generic

type T is private;

Size : integer;

package G\_Queue is

type Queue is limited private;

procedure Enqueue(Q: inout Queue; X:T);

procedure Dequeue(Q: inout Queue; X:T);

procedure Init(Q: out Queue);

procedure Destroy(Q: inout Queue);

function IsFull(Q: Queue);

function IsEmpty(Q: Queue);

–- другие процедуры ...

private

type Queue is record

Left, Right: integer;

body : array(1..Size) of T;

end record;

end G\_Queue;

3. Напишите спецификацию абстрактного типа данных HashTable

(перемешанная таблица, хэш-таблица) на языках Модула-2 и Java (тела

методов и модуль реализации можно опустить).

Язык Модула-2.

DEFINITION MODULE HashTables;

FROM Types IMPORT KeyType, ElementType;

TYPE HashTable;

PROCEDURE Init(VAR T:HashTable);

PROCEDURE Destroy(VAR T:HashTable);

PROCEDURE Lookup(VAR T:HashTable;

Key: KeyType; VAR X:ElementType):BOOLEAN;

PROCEDURE Add(VAR T:HashTable; Key: KeyType; X:ElementType);

PROCDURE Remove(VAR T:HashTable; Key: KeyType):BOOLEAN;

VAR Done: BOOLEAN;

END HashTables.

Язык Java:

interface IHashTable : Iterable

{

ElementType Lookup(KeyType Key);

void Add(KeyType Key, ElementType El);

bool Remove(KeyType Key);

}