

Министерство образования и науки Российской Федерации



Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

**Пермский национальный исследовательский
политехнический университет**

Электротехнический факультет

Кафедра « Информационные технологии и автоматизированные системы »

Верификация программ

ЗАДАНИЯ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ
по дисциплине

ТЕОРИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Пермь, 2016

Задание

Необходимо выполнить комплексное задание, включающее разработку спецификации программы, построение инварианта, построение программы и проверку правильности программы.

Ориентировочная последовательность действий

- построить спецификацию программы
- построить инвариант и ограничивающую функцию
- составить аннотация цикла
- построить программу на метаязыке
- провести проверку правильности построения цикла
- переписать программу на любом языке программирования

Варианты заданий

Таблица 1 - Варианты заданий

Вариант	Задание
1	Реализация умножения через сложение.
2	Поиск первого вхождения элемента в массив.
3	Проверка массива на упорядоченность.
4	Подсчет числа площадок в массиве.
5	Упорядочить значения трех переменных.
6	Определение n-ного числа Фибоначи.
7	Поиск элемента в массиве методом половинного деления.
8	Нахождение ближайшего целого к корню из n.
9	Найти сумму элементов массива.
10	Найти сумму числовой последовательности от a до b
11	Найти сумму положительных элементов массива
12	Найти сумму элементов массива со значением большим 10.
13	Найти сумму элементов массива с четными индексами
14	Определить, все ли четные элементы массива нулевые
15	Вычислить количество нулевых элементов в массиве
16	Заменить в массиве нолями все элементы большие 10

17	Увеличить значение всех элементов массива на единицу
18	Определить, есть ли ненулевые элементы в массиве
19	Проверить, не превышает ли значение каждого из элементов массива числа элементов массива.
20	Определить число элементов массива больших нуля

Состав отчета

- титульная страница
- постановка задачи
- аннотация цикла
- программа на метаязыке
- проверка правильности цикла
- листинги программ

Теоретические сведения

Синтаксис цикла

Традиционный цикл выглядит следующим образом:

```
While B do S
```

В нотации метаязыка вышеприведенная запись переводится в вариант:

```
do B → S od
```

В общем случае вид цикла следующий.

```
do
```

```

    B1 → S1
  ▣ B2 → S2
    . . . .
  ▣ Bn → Sn

```

```
od
```

Повторять следующие действия пока возможно выбрать команду B_i , которая истинна, и выполнить соответствующее S_i .

```
do BB → if
```

```
B1 → S1
```

```
...
```

```
▣ Bn → Sn
```

```
fi
```

```
od
```

Слабейшее предусловие

Введем команду повторения через слабейшее предусловие:

$H_x(R)$ – предикат

Наш цикл завершится за k или менее шагов при истинном R .

$H_0(R) = \neg B \vee R$

Не все охранные условия ложные, R – истина.

$H_k(R) = H_0(R) \vee wp(\text{If}, H_{k-1}(R))_{k>0}$

$H_1(R) = H_0(R) \vee wp(\text{If}, H_0(R)) = H_0(R) \vee wp(\text{If}, \neg B \vee R)$

$wp(\text{DO}, R) = \exists k: 0 \leq k: H_k(R)$

Инвариант

Пример: посчитать сумму элементов массива $b[0:10]$

$b[0:10]$

$i, S := 1, b[0];$

do $i < 11 \rightarrow i, S := i+1, S+b[i]$ od

$\{R: S = (\sum_k b[k])\}$

Сформируем предусловие

Введем предикат $P: 1 \leq i \leq 11 \wedge S = (\sum_k b[k])$

Если P истина перед циклом, перед каждым шагом цикла и после него, то P истинно после цикла.

Нетрудно заметить, что

$P \wedge i \geq 11 \Rightarrow R$

$\{P = \text{True}\}$

$i, S := 1, b[0];$

$\{R\}$ проверяем истинность P

do

$i < 11 \rightarrow \{P \wedge i < 11\} i, S := i+1, S+b[i]$

$\{P\}$

od

$\{i \geq 11 \wedge P\}$ цикл завершился

$\{R\}$

1. P истинно перед циклом

2. P истинно перед каждым шагом

Отсюда вывод: истинность P и ложность охранных условий позволяет заключить, что искомым результатом R получен.

Доказательство:

1. Проверим истинность P после инициализации переменных

$$\begin{aligned} wp("i, S := 1, b[0]", P) &= (1 \leq 1 \leq 11 \wedge b[0] = (\sum_k b[k])) = \\ &= (1 \leq 1 \leq 11 \wedge b[0] = b[0]) = T \end{aligned}$$

2. докажем истинность P после шага цикла, начавшемся при истинном P и истинной охранный.

$$wp("i, S := i+1, S+b[i]", P) =$$

$$= (1 \leq i+1 \leq 11 \wedge S + b[i] = (\sum_k k : 0 \leq k \leq i+1 : b[k])) =$$

$$= (0 \leq i \leq 11 \wedge S = \sum_k k : 0 \leq k \leq i : b[k]) = T$$

3. P может служить предусловием

$$P \wedge i \geq 11 \Rightarrow R$$

$$(1 \leq i \leq 11 \wedge S = (\sum_k k : 0 \leq k \leq i : b[k])) \wedge i \geq 11 \Rightarrow (\sum_k k : 0 \leq k \leq 11 : b[k]) =$$

$$= (i = 11 \wedge S = \sum_k k : 0 \leq k \leq i : b[k]) \Rightarrow$$

$$\Rightarrow S = (\sum_k k : 0 \leq k \leq 11 : b[k]) = (\text{вместо } i \text{ подставляем } 11)$$

видим, левая и правая части равны.

Предикат P истинный перед и после выполнения каждого шага цикла, называется инвариантом цикла.

Введем функцию t, которая покажет, сколько шагов до конца цикла осталось.

$$t := 11 - i$$

Это ограничивающая функция. Показывает верхнюю границу числа оставшихся шагов.

Составим программу, которая $z \leftarrow a * b$ при $b \geq 0$

{b ≥ 0}

x, y, z := a, b, 0;

do

 y > 0 ∧ чет(y) → y, x := y ÷ 2, x + x

 ■ нечет(y) → y, z := y - 1, z + x

od

{R: z = a * b}

Инвариант P: $y \geq 0 \wedge z + x + y = a * b$

1. Проверяем истинность P после инициализации переменных

$$wp("x, y, z := a, b, 0", P) = (b \geq 0) \wedge (0 + a * b = a * b) = (b \geq 0)$$

2. Перед шагом и после шага цикла

а) $z + x * y$	$z + (y/2) * (x + x)$	← Эти скобки раскрывать нельзя, но т.к. y – четное, то можно раскрыть.
до шага	после шага	
б) $z + x * y$	$z + x + y - 1 * x = z + y * x$	
до шага	после шага	

3. Истинность инварианта и ложность охран → постусловие.

$$P \wedge \neg(y > 0 \wedge \text{чет}(y)) \wedge \neg \text{нечет}(y) =$$

$$= P \wedge (y \leq 0) = (y \leq 0 \vee \neg \text{чет}(y) \wedge \text{чет}(y) \wedge P) =$$

$$= (y \leq 0 \wedge \text{чет}(y) \wedge P) = (y \leq 0 \wedge \text{чет}(y) \wedge (y \geq 0 \wedge z + y * x = a * b)) =$$

$$= (y = 0 \wedge z + y * x = a * b \Rightarrow z = a * b)$$

Ограничивающая функция $t = y$

Теорема о цикле, его инварианте и ограничивающей функции

Посылки:

1. $P \wedge B_i \Rightarrow wp(S_i, P)$ - сохранение инварианта

Т.е. если выполняется инвариант и нас пустила i -ая охрана, то выполнение i -ой команды составит истинность P .

2. $P \wedge BV \Rightarrow t > 0$ (т.е. есть еще шаги)

Т.е. если выполняется инвариант и BV , то ограничивающая функция положительна.

3. С каждым шагом ограничивающая функция уменьшается.

$P \wedge B_i \Rightarrow wp("t_1 := t; S_i", t < t_1)$

Т.е. если нас пустила i -ая охрана, то выполнение i -ой команды уменьшит функцию.

Вывод: если эти посылки выполняются, то P можно использовать в качестве предусловия для цикла. $P \Rightarrow wp(DO, P \wedge \neg BV)$;

Аннотирование цикла

Перед циклом

{Q}

{inv P: инвариант}

{bound t: ограничивающая функция}

do

$B_1 \rightarrow S_1$

$\bigwedge B_2 \rightarrow S_2$

...

$\bigwedge B_n \rightarrow S_n$

od

{R : постусловие}

Список условий для проверки цикла

- 1) P истинно перед выполнением цикла
- 2) P является инвариантом цикла $\{P \wedge B_i\} S_i \{P\}$
- 3) Выполнение P и невыполнение BV должно дать R : $P \wedge BV \Rightarrow R$
- 4) Если цикл еще не закончен, то ограничивающая функция положительна:
 $P \wedge BV \Rightarrow t > 0$
- 5) Каждый шаг цикла ведет к концу цикла
 $\{P \wedge B_i\} t_1 := t; S_i \{t < t_1\}$