

ДИСЦИПЛИНА	Методы верификации и валидации (полное наименование дисциплины без сокращений)
ИНСТИТУТ	характеристик программного обеспечения
КАФЕДРА	информационных технологий математического обеспечения и стандартизации (полное наименование кафедры)
ВИД УЧЕБНОГО МАТЕРИАЛА	информационных технологий Материалы для практических/семинарских занятий (в соответствии с пп.1-11)
ПРЕПОДАВАТЕЛЬ	Петренко Александр Анатольевич (фамилия, имя, отчество)
СЕМЕСТР	3, 2024-2025 (указать семестр обучения, учебный год)

Дедуктивная верификация последовательных программ

На основе изучения материала лекций по дисциплине «Методы верификации и валидации характеристик программного обеспечения» требуется выполнить следующее.

1. Докажите следующее условие корректности, вычислив слабое предусловие:

```
{true}
y := 0;
z := a;
y := y + x;
z := z * x;
y := y * z;
z := z / a;
z := z * b;
z := z + c;
y := y + z
{y = a•x2 + b•x + c}
```

2. Докажите следующее условие корректности:

```
{a > 0 & b > 0 & a ≠ b}
x := a;
y := b;
if x > y then
    x := x - y
else
    y := y - x
end
{a > 0 & b > 0 & НОД(x, y) = НОД(a, b)}
```

3. Предложите инварианты циклов для программ P, Q и S, реализующих алгоритм Евклида (см. предыдущее практическое занятие).

4. Докажите частичную корректность реализаций алгоритма Евклида (программы P, Q и S из предыдущего практического занятия).

5. Напишите пред- и постусловие для программ сортировки числовых массивов. Докажите частичную корректность программы, реализующей «метод пузырька».

6. Выполните верификацию следующей программы для заданных пред- и постусловий:

```
{a ≥ 0}
```

```

x := a;
n := 1;
y := 0;
while x ≠ 0 do
    y := y + n;
    n := n + 2;
    x := x - 1
end
{y = a2}

```

7. Выполните верификацию следующей программы для заданных пред- и постусловий:

```

{true}
x := a;
n := 0;
while x ≠ 0 do
    x := x & (x - 1);
    n := n + 1
end
{n = count(a)}

```

Здесь & – операция побитового И, а count(x) — функция, возвращающая число единиц в двоичном представлении числа x. Определите формально предметную область (для определенности можете считать, что переменные принимают целочисленные значения из отрезка [0,232-1]).

8. Выполните верификацию следующей программы для заданных пред- и постусловий:

```

{a > 1}
i := a - 1;
x := 1;
while i > 0 do
    if a % i = 0 then
        k := 0;
        j := i - 1;
        while j > 1 do
            if i % j = 0 then
                k := k + 1
            end;
            j := j - 1
        end;
        if k = 0 then
            x := i;
            i := 1
        end
    end;
end;
end;

```

$i := i - 1$

end

{ $x = \text{maxPrimeFactor}(a)$ }

Здесь $\text{maxPrimeFactor}(a)$ – максимальный простой делитель целого числа a .