МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Воронежский государственный университет

Факультет компьютерных наук

кафедра программирования и информационных технологий

Отчёт

по лабораторной работе № 12

«Разработка критичных систем с использованием формальных методов

Perfect Developer»

по дисциплине

###### «Методы и средства проектирования ИС и технологий»

Выполнил студенты группы 3.1

Мубаракшин Дмитрий

Проверил:

Власов Сергей Васильевич

Воронеж

2014

**Упражнение 1.** Создание проекта.

4. Отсутствие ошибок в спецификации обозначает, что все свойства и предикаты удовлетворены.

5. warning : Cannot prove a < c

Причина: при b > c условие a < c может как выполниться, так и нет.

**Упражнение 2.**

1. const zeroToOneHundred: seq of int ^= 0..100;

property assert 42 in zeroToOneHundred,

101 ~in zeroToOneHundred;

2. function divides(i, j: int): bool

pre j > 0

^= i % j = 0;

3. const squaresOfPrimes: seq of int

^= for those i::2..100

:- forall j::2..<i :- ~divides(i, j)

yield i ^ 2;

4. function max(S: set of int): int

pre ~S.empty

^= that x::S :- forall y::S :- y <= x;

**Упражнение 3.** Ошибки исправлены, также реализована проверка того, что умножение объекта Money на 2 эквивалентно добавлению его к самому себе, и определен оператор «+» без проверки одинаковости валюты, если один из операндов – 0.

*class Currency ^= enum unspecified, euro, pound, USdollar end;*

*final class Money ^=*

*abstract*

*var amt: int,*

*ccy: Currency;*

*invariant amt >= 0,*

*ccy ~= unspecified@Currency | amt = 0;*

*interface*

*function amt, ccy;*

*function worthHaving: bool*

*^= amt >= 0;*

*function plus(m: Money): Money*

*pre m.ccy = ccy*

*^= Money{amt + m.amt, ccy}*

*assert result.ccy = ccy;*

*operator (n: int) \* : Money*

*pre n > 0*

*^= Money{amt \* n, ccy};*

*operator (m: Money) + : Money*

*pre m.amt = 0 | m.ccy = ccy*

*^= Money{amt + m.amt, ccy}*

*assert result.ccy = ccy;*

*property(a, doubled, summed : Money)*

*pre a.ccy ~= unspecified@Currency, doubled = Money{a.amt \* 2, a.ccy}, summed = Money{a.amt + a.amt, a.ccy}*

*assert doubled.amt = summed.amt;*

*nonmember schema swap(a!, b!: Money)*

*pre a.ccy = b.ccy*

*post change a,b satisfy a'=b, b'=a*

*assert a'.ccy = b'.ccy,*

*a.plus(b) = b'.plus(a');*

*schema !inflate(howMuch: int)*

*pre 0 < howMuch < 200*

*post amt! = (amt \* howMuch)/100;*

*build{a: int, c: Currency}*

*pre a >= 0*

*post amt! = a, ccy! = c;*

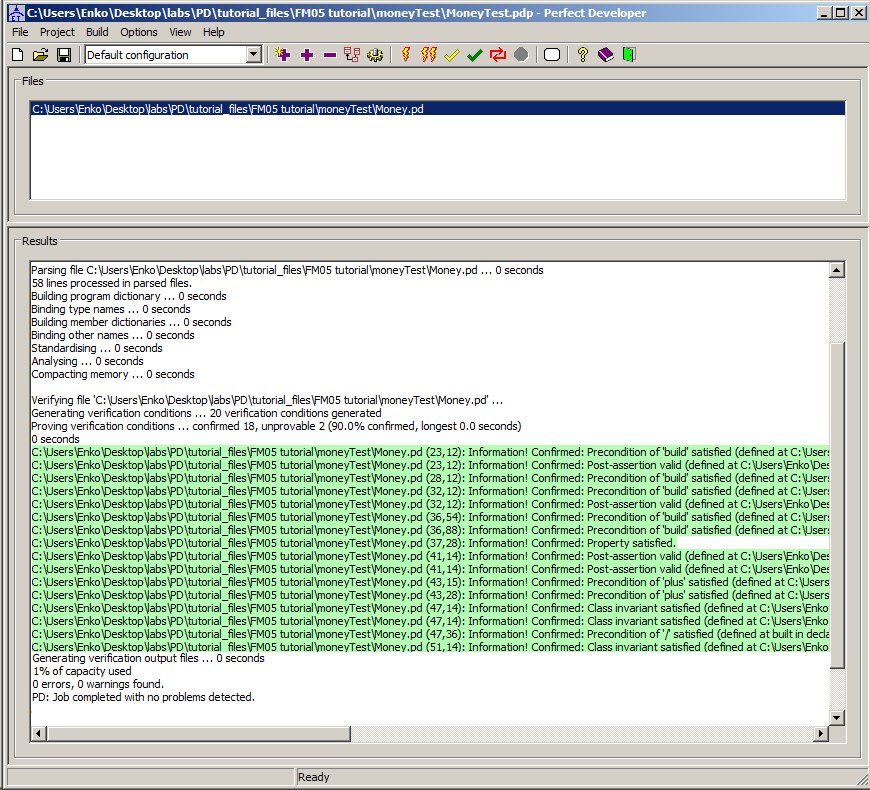
*build{!amt: int}*

*post ccy! = euro@Currency;*

*build{}*

*^= Money{0, unspecified@Currency};*

*end;*



**Упражнение 4.** Генерация графического интерфейса.

