Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по творческому заданию (часть 1)**

Дисциплина: основы алгоритмизации и программирования

Тема: Калькулятор доходности банковских вкладов

Вариант 17

Выполнил:

Студент группы РИС-20-1б

Кузнецов Михаил Сергеевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Постановка задачи**

1) Необходимо разработать программу калькулятор с графическим интерфейсом, которая рассчитывает прибыльность банковских вкладов.

2) Калькулятор получает на вход величину вклада в рублях и копейках или в долларах и центах, а также величину срока в месяцах или годах и вариант условий размещения капитала с вычислением величины вклада к концу указанного срока.

3) Калькулятор должен иметь окна для исходных данных и результатов, для ввода данных, операций и служебных клавиш, клавиши изменения знака числа и вида валюты. Должен позволять корректировать последнее введённое число и выдавать сообщения об ошибках.

**Анализ задачи**

1. Определить какие действия необходимо выполнить для решения задачи.

* Разработать интерфейс программы средствами qt.
* Реализовать поля ввода данных.
* При нажатии кнопки, считать данные, выполнить расчёты, вывести результат и если необходимо сообщения об ошибках

1. Весь алгоритм программы реализован в одном классе MainWindow, унаследованном от стандартного класса QMainWindow.
2. Экземпляры всех элементов интерфейса хранятся в переменной ui.

Ui::MainWindow \*ui;

1. Ввод данных осуществляется через поля QLineEdit, для которых установлены валидаторы, не позволяющие пользователю ввести посторонние символы.

ui->LE\_SumVkl->setValidator(*new* QIntValidator(1, 999999999, *this*));

ui->LE\_CurrencyPart->setValidator(*new* QIntValidator(1, 99, *this*));

ui->LE\_Proc->setValidator(*new* QDoubleValidator(0.0, 100.0, 2, *this*));

ui->LE\_PlacementPeriod->setValidator(*new* QIntValidator(1, 99999, *this*));

1. Выбор срока и периода начисления процентов осуществляется через QComboBox.
2. Считывание размера вклада.

long long int SumVkl = ui->LE\_SumVkl->text().toLongLong() \* 100 + ui->LE\_CurrencyPart->text().toLongLong();

double Sum = SumVkl;

long long int SumVklBuf = SumVkl;

1. Перевод введённого пользователем периода в дни.

int NumDays = 0;

*if* (ui->CB\_PeriodType->currentText() == "Дней") NumDays = ui->LE\_PlacementPeriod->text().toInt();

*else* *if* (ui->CB\_PeriodType->currentText() == "Месяцев") NumDays = ui->LE\_PlacementPeriod->text().toInt() \* 30;

*else* *if* (ui->CB\_PeriodType->currentText() == "Лет") NumDays = ui->LE\_PlacementPeriod->text().toInt() \* 365;

1. Вычисление количества начислений процентов.

int ProcCount = 0;

*if* (ui->CB\_ProcType->currentText() == "В день") ProcCount = NumDays;

*else* *if* (ui->CB\_ProcType->currentText() == "В месяц") ProcCount = NumDays / 30;

*else* *if* (ui->CB\_ProcType->currentText() == "В год") ProcCount = NumDays / 365;

1. Вычисление начисленных процентов.

double proc = ui->LE\_Proc->text().replace(",", ".").toDouble();

proc = proc / 100 + 1;

Sum \*= pow(proc, ProcCount);

**Функция main()**

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(*argc*, argv);

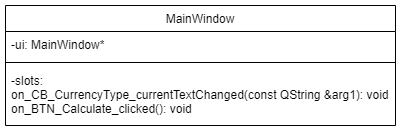
MainWindow w;

w.show();

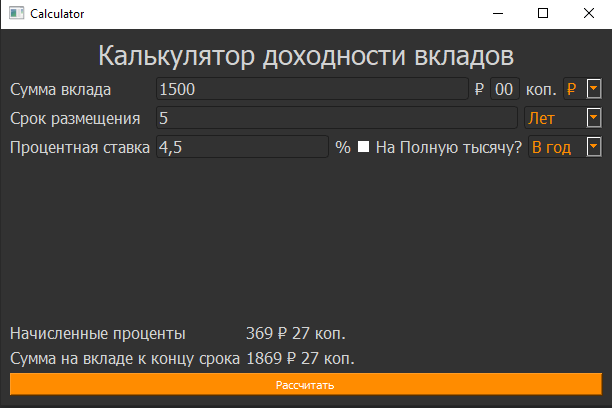
*return* a.exec();

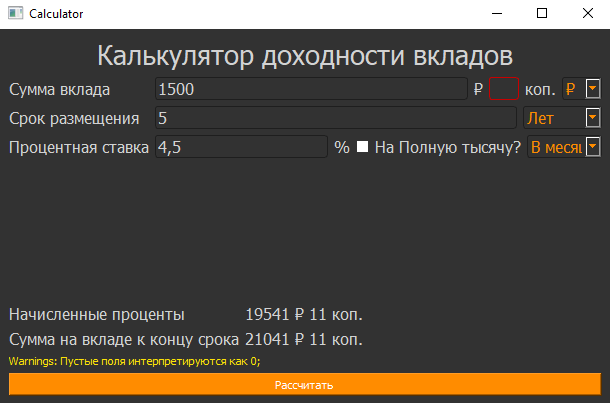
}

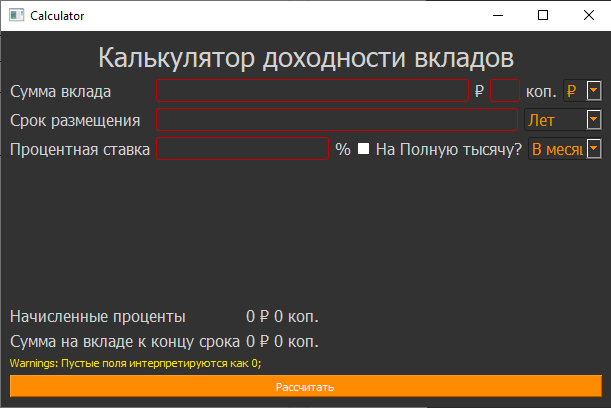
**UML диаграмма**

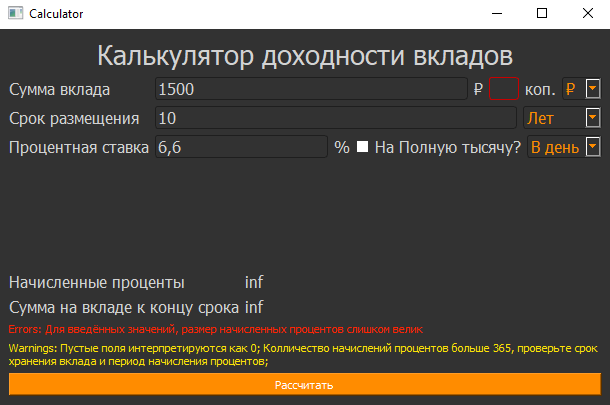
****

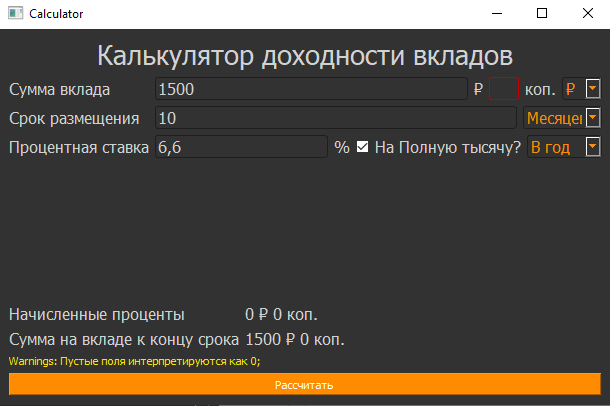
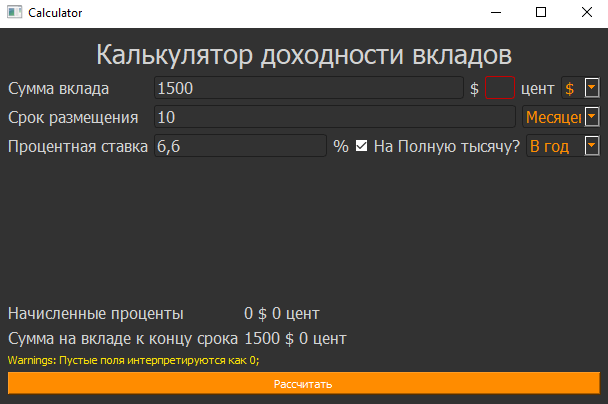
**Скриншоты тестов**

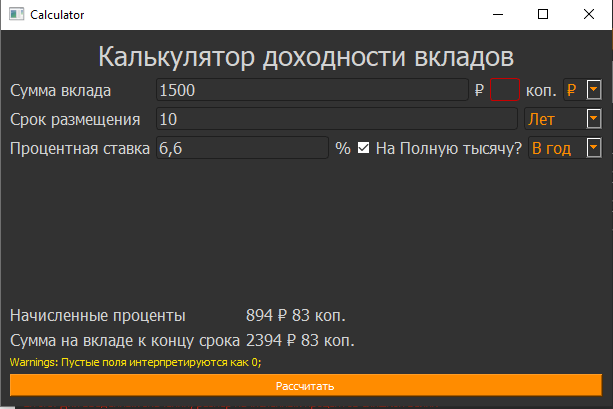
****

****

****

****

**** ****

****

**Объяснение результатов работы**

1. Пустые поля выделяются красным цветом для привлечения внимания и интерпретируются программой как 0, поэтому ошибки не вызывают.
2. Единственная возможная ошибка возникает если вычисленный результат оказывается слишком большим.
3. Если пользователь выставил не корректный срок и период начисления процентов (например, срок – 10лет, а период начисления – ежедневный), программа выводит предупреждение о том, что количество начислений процентов слишком велико.