Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Иркутский национальный исследовательский технический

университет»

Институт информационных технологий и анализа данных

**О Т Ч Ё Т**

|  |  |  |
| --- | --- | --- |
| о прохождении | | учебной практики |
|  | | (вид практики: учебная/производственная) |
| технологическая (проектно-технологическая) практика | | |
| (тип практики: технологическая/научно-исследовательская работа/преддипломная и др.) | | |
|  | | |
| в | Институте ИТиАД | |
|  | (наименование профильной организации) | |

Обучающегося

(ФИО, группа, подпись)

Руководитель практики от института ИТиАД

Осипова Е.А., доцент ЦПИ

(ФИО, должность, подпись)

Руководитель образовательной программы

Аршинский В.Л., доцент ЦПИ

(ФИО, должность, подпись)

Оценка по практике

(ФИО, подпись, дата)

Содержание отчета на \_\_\_ стр. Приложение к отчету на \_\_\_ стр.

Иркутск 2023

**Индивидуальное задание на прохождение**

**учебной практики: технологическая (проектно-технологическая) практика**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| для |  | | | | |
|  | (ФИО обучающегося полностью) | | | | |
| обучающегося | | 1 | курса | группы |  |

по направлению подготовки Информационные системы и технологии

профиль Информационные системы и технологии в административном управлении

Место прохождения практики: Институт ИТиАД

Сроки прохождения практики с «\_\_» \_\_\_\_\_ 2023 г. по «\_\_» \_\_\_\_\_ 2023 г.

Цели и задачи прохождения практики:

Содержание практики, вопросы, подлежащие изучению:

Планируемые результаты практики:

Руководитель практики от института ИТиАД

\_\_\_\_\_\_\_\_\_\_\_/ Осипова Е.А. /

(подпись)

**Согласовано:**

Руководитель ООП

\_\_\_\_\_\_\_\_\_\_\_/Аршинский В.Л./

(подпись)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2023 г.

С настоящим индивидуальным заданием и с программой практики ознакомлен(а), задание принято к исполнению

\_\_\_\_\_\_\_\_\_\_«\_\_\_»\_\_\_\_\_\_\_\_2023 г.

(подпись)

**Содержание**

Введение

Целью практики является совершенствование навыков алгоритмизации, тестирования и программирования на языке высокого уровня, а также закрепление знаний, полученных при изучении дисциплин «Информатика» и «Программирование».

Основными задачами учебной практики являются приобретение практических навыков:

* рационального использования рабочего времени;
* культуры труда;
* разработки алгоритмов решения задач;
* использования различных методов тестирования алгоритмов;
* применения языка программирования высокого уровня;
* качественного выполнения заданий;
* межличностных отношений при коллективном выполнении задания.

1 Постановка задачи

Индивидуальное задание:

Зарплата официантов складывается из оклада и премии. Величина оклада фиксированная и не зависит от прибыли ресторана. Размер премии зависит от величины премиального фонда и количества сотрудников. Премиальный фонд формируется пз чистой прибыли после вычета доли владельца бизнеса (25%).

Реализовать программу для расчета зарплаты официантов.

2 Таблица спецификаций

Таблица 1 – Таблица спецификаций

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Имя** | **Назначение** | **Тип** | **ОДЗ** |
| **int main() – точка входа в программу** | | | | |
| 1 | totalIncome | Входные данные — чистая прибыль | double | [5.0 E-38…5,0E+38] |
| 2 | employCount | Входные данные — количество сотрудников | short | [1…1000] |
| 3 | employIncome | Входные данные — оклад сотрудника | double | (0…5,0E+38] |
| **double calcEmploySalary() – функция для расчета зарплаты одного сотрудника** | | | | |
| 4 | extraEmploy | размер премии | double | [5.0 E-38…5,0E+38] |
| 5 | salaryEmploy | выходные данные — размер зарплаты сотрудника | double | [5.0 E-38…5,0E+38] |

# **3** **Блок-схема алгоритма решения задачи**

На рисунке 1 представлена блок-схема алгоритма решения задачи.

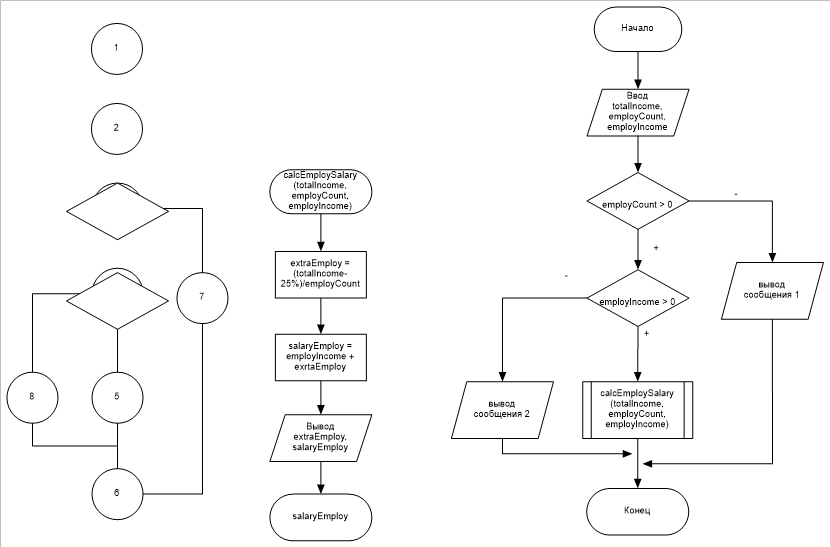


Рисунок 1 – Блок-схема алгоритма решения задания

Данный алгоритм содержит подалгоритм для расчета зарплаты одного сотрудника (см. рис.2)

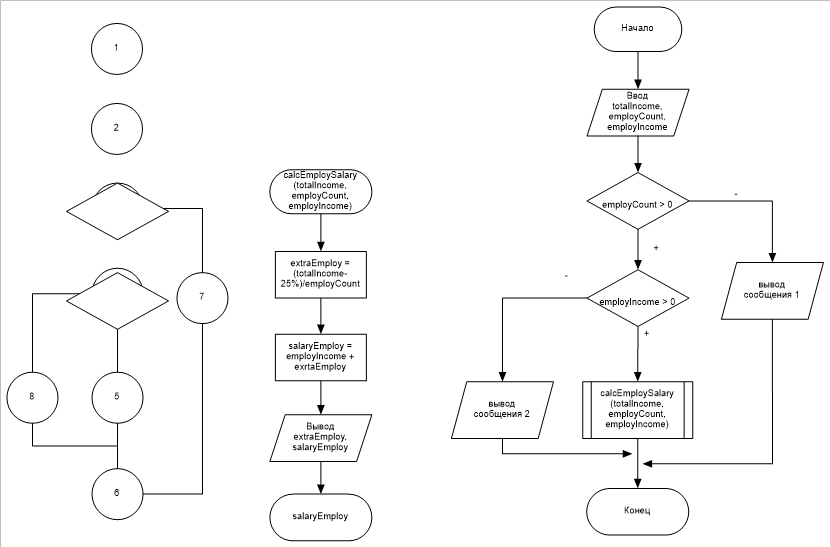


Рисунок 2 – Блок-схема подалгоритма расчета зарплаты на одного сотрудника

# **4** **Таблица тестов**

В соответствии с рекомендованной стратегией тестирования были определены правильные и неправильные классы эквивалентности для входных данных на основе анализа постановки задачи (см. таблицу 2).

Таблица 2 – Выделенные классы эквивалентности

|  |  |  |
| --- | --- | --- |
| № | Правильные классы эквивалентности | Неправильные классы эквивалентности |
| 1 | employCount >0 (№1) | employCount <=0 (№2) |
| 2 | employIncome>0 (№3) | employIncome<=0 (№4) |
|  |  |  |

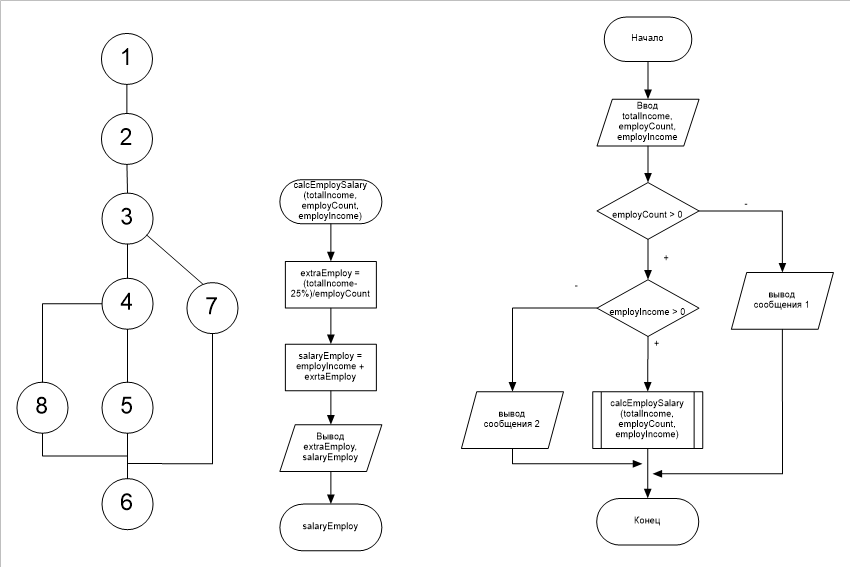
На основе классов эквивалентности были разработаны тесты, представленные в таблице 3. Следует отметить, что в силу особенностей постановки задачи не на каждый правильный класс эквивалентности получился отдельный тест.

Далее в таблицу были добавлены тесты полученные методом граничных значений для каждого правильного класса эквивалентности

Таблица 3 – Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание | Входные данные | Выходные данные |
| 1 | Штатный тест (правильный классы эквивалентности №1, №3) | totalIncome = 200000  employCount = 3  employIncome = 10000 | extraEmploy = 50000  salaryEmploy = 60000 |
| 2 | Отрицательное количество работников (неправильный класс эквивалентности №2) | totalIncome = 200000  employCount = -3  employIncome = 10000 | Сообщение «Количество сотрудников должно быть больше 0» |
| 3 | Отрицательный оклад работников (неправильный класс эквивалентности №4) | totalIncome = 200000  employCount = 3  employIncome = -10000 | Сообщение «Оклад сотрудников должен быть больше 0» |

Чтобы провести тестирование по модели **белого ящика** построим граф потока управления на основе блок-схемы алгоритма. Каждая инструкция представлена отдельным узлом, а ребра показывают направление выполнения алгоритма. Граф потока управления приведен на рисунке 2.

Рисунок 2 — Граф потока управления для алгоритма

В результате выполнения теста 1 выполняется проход по пути *1-2-3-4-5-6*. В результате выполнения теста 2 — путь *1-2-3-7-6*. Теста 3 — *1-2-3-4-8-6*. Имеющийся набор тестов обеспечивает покрытие всех возможных путей, поэтому формировать дополнительные тесты не требуется.

**5 Исходный код**

В исходный алгоритм были внесены изменения для того, чтобы учесть проблемные ситуации выявленные в ходе разработки тестов.

Алгоритм был реализован на языке программирования C, код доступен по ссылке <https://repl.it/@dpprosk/exapleforsummerpractice>

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main(void) {

float x, y, a, b, c;

printf("Введите x ");

if (scanf("%f", &x) == 0){ //если функция получает неподходящие данные (в нашем случае строку вместо чисел), то она вернет 0

printf("Ошибка! Нужно ввести число.");

return 1;

}

printf("Введите a ");

scanf("%f",&a);

printf("Введите b ");

scanf("%f",&b);

printf("Введите c ");

scanf("%f",&c);

if ((x - 1 < 0) && (b - x != 0)) {

y = a\*(pow(x, 2)) + b;

}

else {

if ((x - 1 > 0) && (b + x == 0)) {

y = (x - a)/x;

}

else {

if (c != 0) {

y = x/c;

}

else {

printf("Ошибка! Деление на ноль.");

return 1;

}

}

}

printf("y = %f", y);

return 0;

}

**6 Анализ вычислительной сложности алгоритма**

Выполним асимптотическую оценку функции сложности алгоритма.

Опишем алгоритм в виде псевдокода:

1. float x, y, a, b, c

2. print(«Введите x»)

3. read(x)

4. if (type(x) == String) {

5. print(«Ошибка! Нужно ввести число.»)

6. return 1}

7. print(«Введите a»)

8. read(a)

9. print(«Введите b»)

10. read(b)

11. print(«Введите c»)

12. read(c)

13. if (x — 1 < 0) И (b — x != 0) {

14. y = a\*(x^2) + b}

15. else {

16. if (x — 1 > 0) И (b + x == 0) {

17. y = (x — a)/x}

18. else {

19. if (c != 0) {

20. y = x/c }

21. else {

22. print(«Ошибка! Деление на ноль.»)}}}

23. print(y)

24. return 0

Проанализировав псевдокод, видим, что размер входных данных для алгоритма является постоянным, следовательно, алгоритм не зависит от объема входных данных. В результате выполнения алгоритма каждый из операторов 1-24 выполняется не более одного раза. Исходя из вышесказанного, можно заключить, что алгоритм имеет постоянное время выполнения. Используя О-нотацию, верхняя асимптотическая граница функции сложности исследуемого алгоритма записывается как **O(1)**.

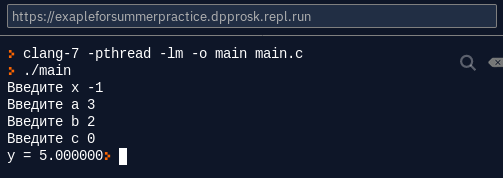
**7 Результаты тестирования**

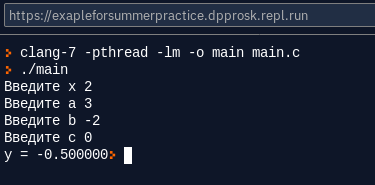
В таблице 4 приведены результаты тестов из таблицы 3.

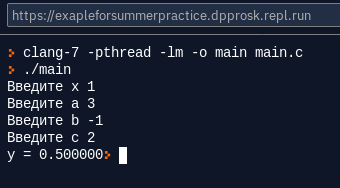
Таблица 4 – Таблица результатов тестирования

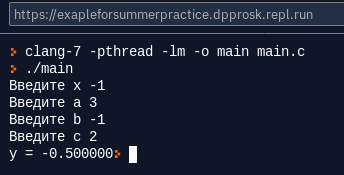
|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Выходные данные | Результат |
| 1 | x = -1, b = 2, a = 3, c = 0 | 5 | Тест пройден |
| 2 | x = 2, b= -2, a = 3, c =0 | -0.5 | Тест пройден |
| 3 | x = 1, c = 2, b = -1, a = 3 | 0.5 | Тест пройден |
| 4 | x = -1, b = -1, c = 2, a = 3 | -0.5 | Тест пройден |
| 5 | x = 2, b = 2, c = 2, a = 3 | 1 | Тест пройден |
| 6 | x = 1, c = 0, a = 3, b = 2 | «Ошибка! Деление на ноль.» | Тест пройден |
| 7 | x = -1, b = -1, c = 0, a = 3 | «Ошибка! Деление на ноль.» | Тест пройден |
| 8 | x = 2, b = 2, c = 0, a = 3 | «Ошибка! Деление на ноль.» | Тест пройден |
| 9 | x = «asdfg», a = 1, b = 1, c = 1 | «Ошибка! Нужно ввести число.» | Тест пройден |

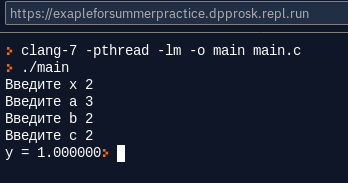
Результаты выполнения тестов с 1 по 9 приведены на рисунках с 3 по 12 соответственно.

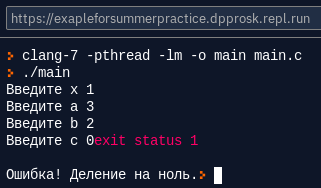
Рисунок 3 – Результат выполнения теста 1

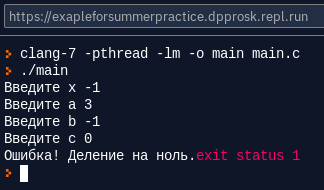
Рисунок 4 – Результат выполнения теста 2

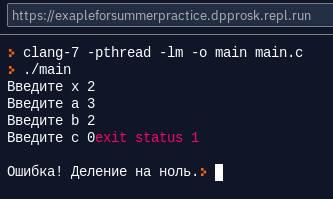
Рисунок 5 – Результат выполнения теста 3

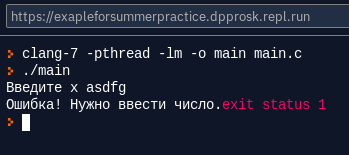
Рисунок 6 – Результат выполнения теста 4

Рисунок 7 – Результат выполнения теста 5

Рисунок 8 – Результат выполнения теста 6

Рисунок 9 – Результат выполнения теста 7

Рисунок 10 – Результат выполнения теста 8

Рисунок 11 – Результат выполнения теста 9

**Заключение**

В процессе практики был разработан алгоритм решения индивидуального задания: «вычисление значения функции y в зависимости от значения аргумента x и коэффициентов a, b, c». С использованием методов классов эквивалентности и предположения об ошибке сформировано 9 тестов, покрывающих все пути. Составленная на языке Си программа, реализующая разработанный алгоритм, прошла все 9 тестов успешно.

В ходе учебной практики были получены навыки составления блок-схем алгоритма решения задачи, разработки тестов с использованием различных методов, а также применения языка программирования Си для реализации алгоритма решения задачи.

Все цели практики достигнуты.

**Список литературы**

1. Павловская Т.А. С и С++. Программирование на языке высокого уровня. СПб.: Питер, 2003. – 461с.
2. Программирование на языке Си [Электронный ресурс] URL: http://www.codenet.ru/progr/cpp/1/ (дата обращения: 29.06.20)