

Белорусский государственный технологический университет

Факультет информационных технологий

Специальность «Программная инженерия»

Лабораторная работа 6

По дисциплине «Основы программной инженерии»

На тему «Модульное программирование»

Выполнил:

Студент 1 курса 10 группы, 1 подгруппы

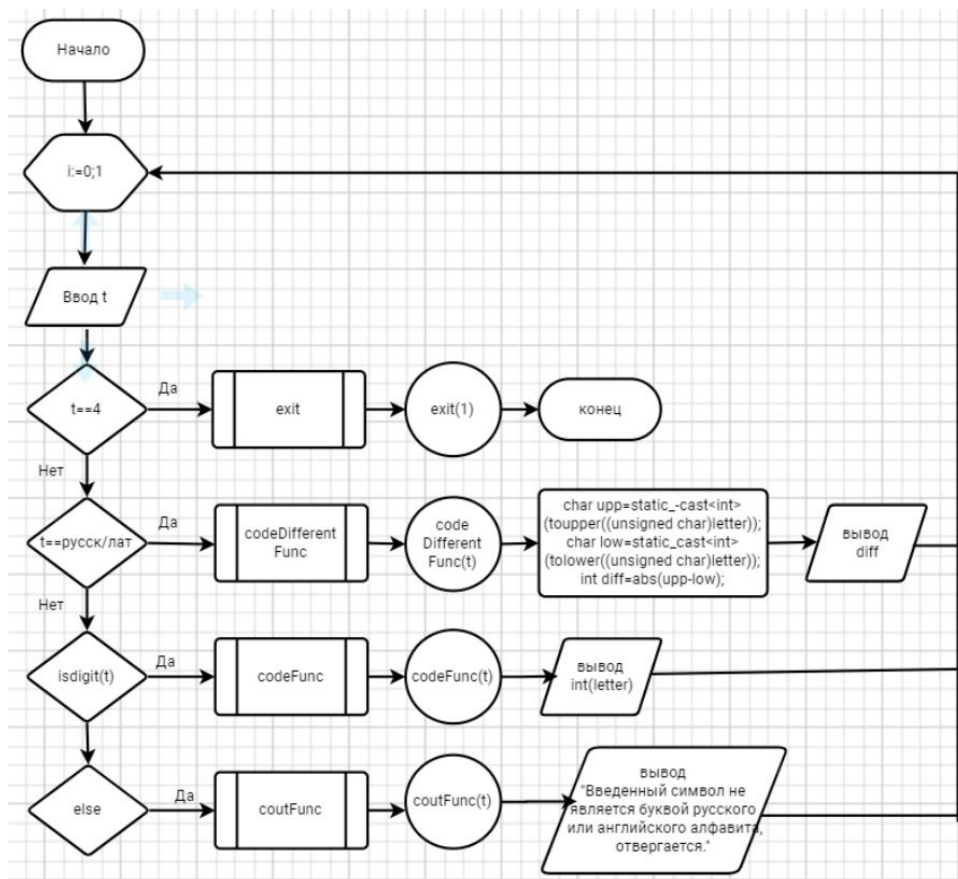
Пронович Максим Сергеевич

2023, Минск

2. Входные данные: t.

Выходные данные: diff_code, Compound.

3. Блок-схема в модульном программировании:



Разбор программы по модулям модули (screenshot 1/2/3/4 – использование в коде/заголовочный файл/файл реализации модуля), состав, назначение, входные/выходные данные, алгоритм:

1:

```
{
    CodeDifferentFunc(t);
}
```

```
int CodeDifferentFunc(char Compound)
{
    char to_upper = static_cast<int>(toupper((unsigned char)Compound));
    char to_low = static_cast<int>(tolower((unsigned char)Compound));
    int diff_code = abs(to_upper - to_low);
    cout << "Разница в значениях кодов: " << diff_code << endl;
    return diff_code;
}
```

Состав: Compound.

Входные данные: t

Выходные данные: diff_code

Назначение: упрощение исходного кода, упрощение тестирования программы и анализа ошибок.

2:

```
{  
    CodeFunc(t);  
}
```

```
int CodeFunc(char Compound)  
{  
    cout << Compound << " " << "ascii код: " << int(Compound) << endl;  
    return Compound;  
}
```

Состав: Compound

Входные данные: t

Выходные данные: Compound

Назначение: упрощение исходного кода, упрощение тестирования программы и анализа ошибок.

3:

```
{  
    CoutFunc(t);  
}
```

```
void CoutFunc(char Compound)  
{  
    cout << "Введенный символ не является буквой русского или латинского алфавита" << endl;  
}
```

Состав: Compound

Входные данные: t

Выходные данные: Строка

Назначение: упрощение исходного кода, упрощение тестирования программы и анализа ошибок.

4:

```
if (t == '4')  
{  
    exit(i);  
}
```

```
void exit(char Compound)  
{  
    Compound = 1;  
}
```

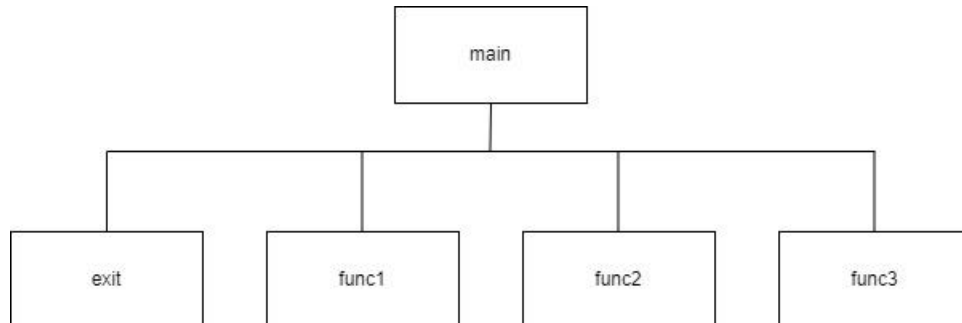
Состав: Compound

Входные данные: i

Выходные данные: Compound

Назначение: упрощение исходного кода, упрощение тестирования программы и анализа ошибок.

Нисходящее проектирование программы



Псевдокод

ВЫВОД "Для выхода из программы введите 4"

ЦИКЛ ОТ $i := \langle 0 \rangle$ ДО $\langle 1 \rangle$ [0]

ВВОД символа

ЕСЛИ $t == '4'$

exit(i) => завершение программы

ИНАЧЕ ЕСЛИ $t ==$ (латин/русс буква)

func2(t) => Вывод разницы значений кодов

ИНАЧЕ ЕСЛИ $t ==$ цифра

func3(t) => Вывод ascii кода t

ИНАЧЕ ВЫВОД "Введенный символ не является буквой
русского или английского алфавита, отвергается."

RETURN 0