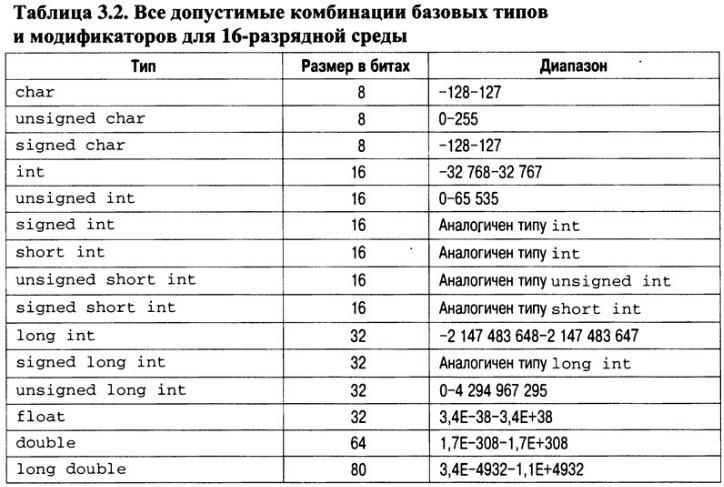
Арсентьевой Н. В. ИВТ - 21

# 1 лабораторная работа: АТД.



АТД Время

* **Данные**

Часы = h

Минуты = m

Секунды = s

* **Операции**
* Конструктор

Начальные значения: -

Процесс:

h = 0

m = 0

s = 0

* Операция ЗадатьЧас:
* Вход: h
* Предусловия: h >= 0 и h <= 23
* Процесс: h = h1
* Выход: -
* Постусловия: -
* Операция ПрочитатьЧас:
* Вход: -
* Предусловия: -
* Процесс: чтение часа
* Выход: h
* Постусловия: -
* Операция ЗадатьСекунды:
* Вход: s
* Предусловия: s >= 0 и s <= 59
* Процесс: s = s1
* Выход: -
* Постусловия: -
* Операция ПрочитатьСекунды:
* Вход: -
* Предусловия: -
* Процесс: чтение секунд
* Выход: s
* Постусловия: -
* Операция ЗадатьМинуты:
* Вход: m
* Предусловия: m >= 0 и m<= 59
* Процесс: m = m1
* Выход: -
* Постусловия: -
* Операция ПрочитатьМинуты:
* Вход: -
* Предусловия: -
* Процесс: чтение минут
* Выход: m
* Постусловия: -
* Операция ПереводВстроку:
* Вход: -
* Предусловия:
* Процесс: Создает строку со значениями
* Выход: строка со значениями
* Постусловия: -
* Операция ВывестиВтекстовыйФайл:
* Вход: -
* Предусловия: -
* Процесс: запись в файл
* Выход: текстовый файл со временем
* Постусловия: -

Конец Время;

## Реализация АТД на C++:

### Заголовочный файл с классом:

/// Класс Время

class Time

{

// Закрытые поля часов, минут, секунд

private:

int h, m, s;

// Методы

public:

Time(); // Конструктор по умолчанию

Time(int h1, int m1, int s1); // Конструктор с параметрами

/// Сеттеры принимающие значения полей

void set\_h(int h1);

void set\_m(int m1);

void set\_s(int s1);

/// Геттеры возращающие значения полей

int get\_h() const;

int get\_m() const;

int get\_s() const;

/// Перевод значений в строку

std::string to\_str() const;

/// Вывод в текстовый файл

void output\_txtfile(const string& filename, Time\* a) const;

};

### Реализация методов в .cpp файле:

/// Реализация класса Время

/// Конструктор по умолчанию

Time::Time() {

h = 0;

m = 0;

s = 0;

}

/// Конструктор с параметрами

Time::Time(int h1, int m1, int s1) {

set\_h(h1);

set\_m(m1);

set\_s(s1);

}

/// Задание часов

void Time::set\_h(int h1) {

if (h1 < 0 || h1 > 23) throw std::invalid\_argument("Час введен некорректно");

else h = h1;

}

/// Задание секунд

void Time::set\_m(int m1) {

if (m1 < 0 || m1 > 59) throw std::invalid\_argument("Минуты введены некорректно");

else m = m1;

}

/// Задание минут

void Time::set\_s(int s1) {

if (s1 < 0 || s1 > 59) throw std::invalid\_argument("Секунды введены некорректно");

else s = s1;

}

/// Вернуть часы

int Time::get\_h() const { return h; }

/// Вернуть минуты

int Time::get\_m() const { return m; }

/// Вернуть секунды

int Time::get\_s() const { return s; }

/// Перевод значений в строку

std::string Time::to\_str() const {

// Множества чисел, от которых зависит написание окончания в словах: час, секунда, минута

set<int> set1 = { 1, 21, 31, 41, 51 };

set<int> set2 = { 2, 3, 4, 22, 23, 24, 32, 33, 34, 42, 43, 44, 52, 53, 54 };

string str = ""; // Пустая строка

/// Проверка для слова час

if (set1.find(h) != set1.end()) { str = str + to\_string(h) + " час "; }

else if (set2.find(h) != set2.end()) { str = str + to\_string(h) + " часа "; }

else { str = str + to\_string(h) + " часов "; }

/// Проверка для слова минута

if (set1.find(m) != set1.end()) { str = str + to\_string(m) + " минута "; }

else if (set2.find(m) != set2.end()) { str = str + to\_string(m) + " минуты "; }

else { str = str + to\_string(m) + " минут "; }

/// Проверка для слова секунда

if (set1.find(s) != set1.end()) { str = str + to\_string(s) + " секунда"; }

else if (set2.find(s) != set2.end()) { str = str + to\_string(s) + " секунды"; }

else { str = str + to\_string(s) + " секунд"; }

return str; // Результат строки

};

/// Вывод в текстовый файл

void Time::output\_txtfile(const string& filename, Time\* a) const {

ofstream f(filename); // Создание и запись в файл

f << "Время: ";

f << a->to\_str(); // Вызов метода преобразования значений полей в строку и вывод результата в текстовый файл

f << endl;

f.close();

};

### Главная программа:

int main()

{

setlocale(LC\_ALL, "ru"); // Подключение русского языка

/// protected code

try

{ const string filename = "time"; // Название текстового файла

// Создание 1 объекта

Time \*object1 = new Time();

// Задание значений часы, минуты, секунды

object1->set\_h(21);

object1->set\_m(59);

object1->set\_s(33);

// Вывод в консоль значений этого объекта

cout << object1->to\_str() << endl;

// Вывод значений в текстовый файл

object1->output\_txtfile(filename, object1);

delete object1; // Очистка памяти

}

catch (std::invalid\_argument &e) // Ловит исключения и выводит их на экран

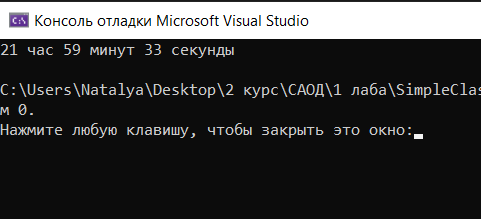
{

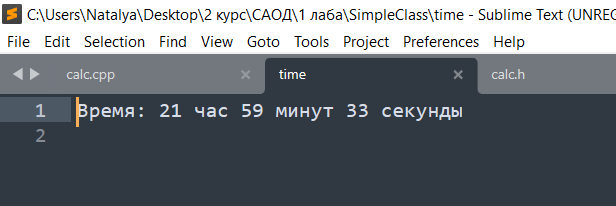
std::cout << e.what();

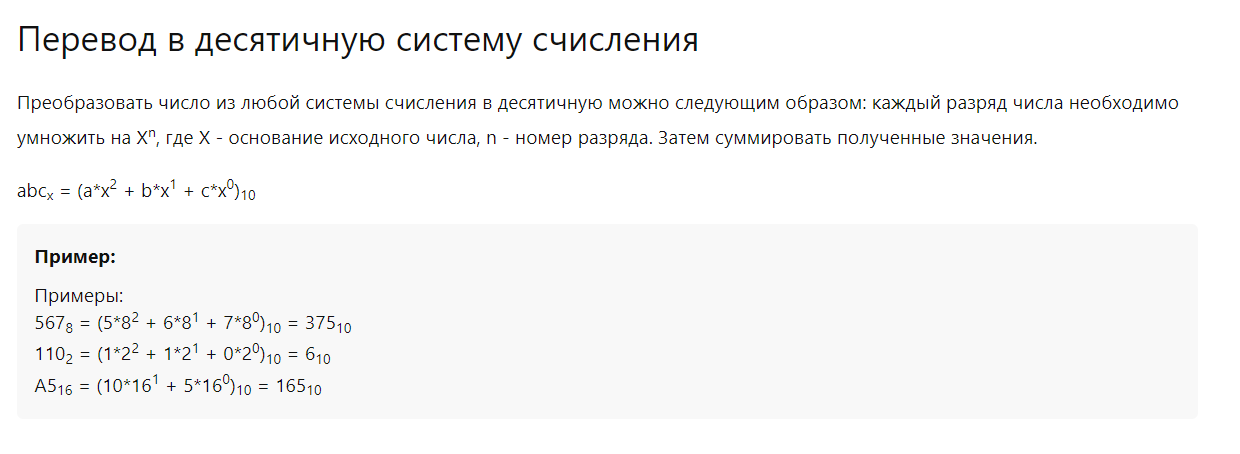
}

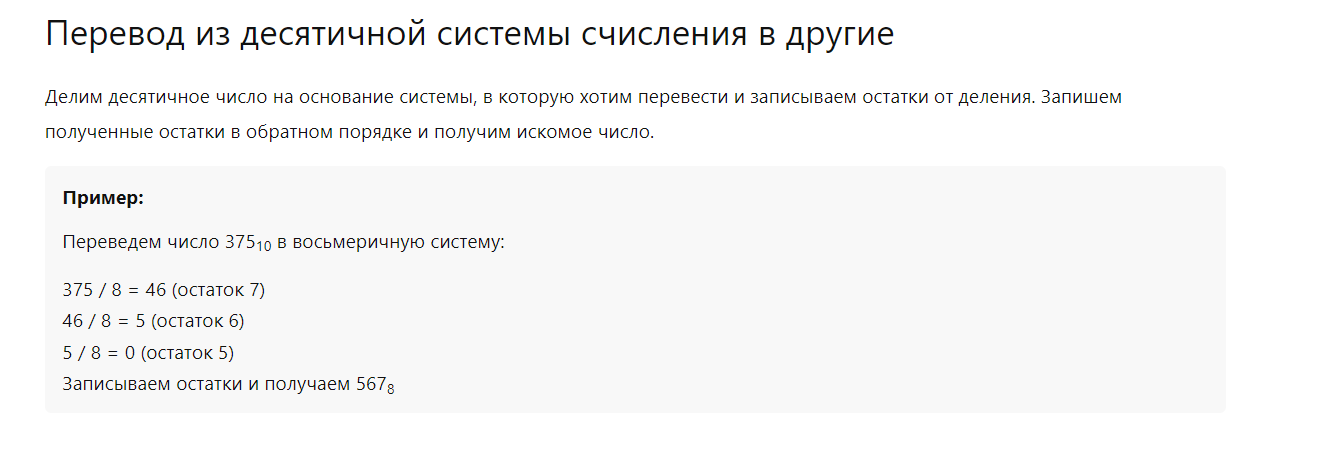
}

## Результат работы класса:









Источники:

<https://studfile.net/preview/6304069/page:3/>

<https://calcus.ru/perevod-sistem-schisleniya>