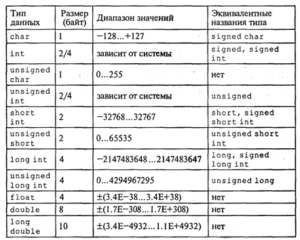
# Типы данных



# Код программы (класс и main функция)

#include <iostream> // потоковый ввод вывод в консоль

#include <exception> // исключения

#include <string> // строки

#include <fstream> // файловый поток

#include <sstream> // строковый поток

using namespace std; // пространство имен для работы со строками и консолью

class itemPricement // Класс описывающий стоимость товара

{

private:

string itemName = ""; // название товара

float itemPrice = 0.0; // его цена

int itemCount = 0; // его количество

public:

itemPricement () // конструктор без параметров

{

}

void setName (string & s1) // сеттер названия

{

if (s1 == "") // если строка пустая то

throw exception ("Name invalid"); // бросим исключение

this->itemName = s1;

if ((s1)!=getName()) throw exception("Set failed"); // ПУ - если значение не присвоилось (если текущее значение НЕ РАВНО присвоенному строкой выше то except)

}

string const getName () // геттер названия

{

return this->itemName;

}

void setPrice (float value) // сеттер цены

{

if ((value < 0) || (value == NULL)) // если ссылка на значение или значение меньше 0

throw exception ("Price invalid"); // бросим исключение

itemPrice = value;

if ((value)!=getPrice()) throw exception("Set failed"); // ПУ - если значение не присвоилось (если текущее значение НЕ РАВНО присвоенному строкой выше то except)

}

float const getPrice () // геттер цены

{

return this->itemPrice;

}

void setCount (int value) // сеттер количества

{

if ((value <= 0) || (value == NULL)) // если ссылка на значение или значение меньше или равно 0

throw exception ("Count invalid");

itemCount = value;

if ((value)!=getCount()) throw exception("Set failed"); // ПУ - если значение не присвоилось (если текущее значение НЕ РАВНО присвоенному строкой выше то except)

}

int const getCount () // геттер количества

{

return this->itemCount;

}

float calcChange (float value) // расчет сдачи

{

if ((value - itemPrice) > 0.01) // если сдача не отрицательная

return value - (itemPrice\*itemCount); // вернем сдачу

}

string const ToString () // метод для вывода класса в виде текстовых данных

{

ostringstream ost;

ost << "Name of item: " << this->itemName

<< "\n" << "Item count: " << this->itemCount

<< "\n" << "Price of 1 item: " << this->itemPrice

<< "$\n" << "Final price: " << (this->getPrice()) \*

(this->getCount()) << '\n';

string s = ost.str ();

return s;

}

void binaryOut(const string &s1) // метод для вывода класса в бинарный файл

{

ofstream fout;

fout.open(s1, ofstream::trunc);

if (!fout.is\_open()) { throw exception(); }

else

std::cout << "File out is opened!" << endl;

/// вывод класса в бинарном виде

fout.write((char\*)&(this->itemName), sizeof(string));

fout.write((char\*)&(this->itemPrice), sizeof(float));

fout.write((char\*)&(this->itemCount), sizeof(int));

fout.close();

}

void textOut(const string &s1) // метод для вывода класса в текстовый файл

{

ofstream fout; // создадим объект файлового потока

fout.open(s1, ofstream::trunc); // откроем его и проверим открытие

if (!fout.is\_open()) { throw exception(); } // иначе бросим исключение

else

std::cout << "File out is opened!" << endl;

fout<<ToString(); // выведем текстовые данные

}

};

int main ()

{

itemPricement \*item = new itemPricement; // инициализировали объект

string value = ""; // инициализация значений которые потом присвоим полям

int value1 = 0;

float value2 = 0;

float finalprice=0.0;

cout << "Enter name of item\n";

cin >> value;

item->setName (value); // сетнули название

cout << "Enter count of item\n";

cin >> value1;

item->setCount (value1); // сетнули количество

cout << "Enter price of item\n";

cin >> value2;

item->setPrice (value2); // сетнули цену за 1 шт

finalprice=item->getPrice()\*item->getCount(); // высчитаем итоговую цену (цена\*количество товара)

cout << "Final price is: " << finalprice << "$\n"; // выведем итог цену

while (true) // бесконечный цикл

{

float temp = 0;

cout << "Input your money: \n"; // юзер ввел деньги свои

cin >> temp;

if (temp >= finalprice) // если их достаточно или даже больше

{

value2 = temp; // вернем в основную программу деньги пользователя

break; // закончим бесконечный цикл

}

else

cout << "Not enough money\n\n";

}

if (value2 > finalprice) // если пользователь дал больше денег чем надо

{

cout << "There are too much money, take your change:\n";

cout << item->calcChange (value2) << "$\n\n"; // отдадим ему сдачу

}

/// выведем отчеты сначала в бинарный файл, затем в текстовый

item->binaryOut("Documents/git work/georgy\_purtov2022/SAOD1/itemPricementBinary.txt");

item->textOut("Documents/git work/georgy\_purtov2022/SAOD1/itemPricement.txt");

// для меня (нюансы разработки на двух ОС)

// fout.open ("Documents/git work/georgy\_purtov2022/SAOD1/itemPricement.txt");

// std::ofstream f("Documents/git work/georgy\_purtov2022/SAOD1/itemPricement.txt");

cout << "\t\t\tThanks for purchasing!";

delete item; // очистим память; файловые потоки закрываются классом

}

# АТД Стоимость товара

• Данные

◦ строковое Название

◦ вещественное Цена

◦ целое Количество

• Операции

◦ Конструктор

Начальные значения:

Процесс:

Имя товара = “”;

Цена= 0;

Количество = 0;

* Конструктор 2

Начальные значения: value1, value2, value3

Процесс:

Имя товара = value1;

Цена= value2;

Количество = value3;

Операция ЗадатьИмя

• Вход: строчная величина

• Предусловия: строчная величина не пустая

• Процесс: Название = строчная величина;

• Выход: -

• Постусловия: название изменилось

Операция ПрочитатьИмя

• Вход:

• Предусловия:

• Процесс: -

• Выход: Название

• Постусловия: -

Операция ЗадатьЦена

• Вход: вещественная величина

• Предусловия: вещественная величина не меньше 0; целочисленная величина не NULL

• Процесс: Цена = вещественная величина;

• Выход: -

• Постусловия: цена изменилась

Операция ПрочитатьЦена

• Вход:

• Предусловия:

• Процесс:

• Выход: Цена

• Постусловия: -

Операция ЗадатьКоличество

• Вход: целая величина

• Предусловия: целая величина не пустая, целая величина>0

• Процесс: количество = целая величина;

• Выход: -

• Постусловия: количество изменилось

Операция ПрочитатьКоличество

• Вход:

• Предусловия:

• Процесс:

• Выход: Количество товара

• Постусловия: -

Операция ВысчитатьСдачу

• Вход: вещественная величина

• Предусловия: вещественная величина не пустая

• Процесс: результат = сдача- цена

• Выход: результат

• Постусловия:

Конец Стоимость товара

# Перевод из 10 СС в двоичную (С++)

int dec2bin(int num)

{

int bin = 0, k = 1;

while (num)

{

bin += (num % 2) \* k;

k \*= 10;

num /= 2;

}

return bin;

}