Universidad Politecnica de Puerto Rico

Asignacion 10.1

Jorge A. Serrano

#121260

CECS 2222 Computer Programming II

Prof. Claudia Talavera

UML

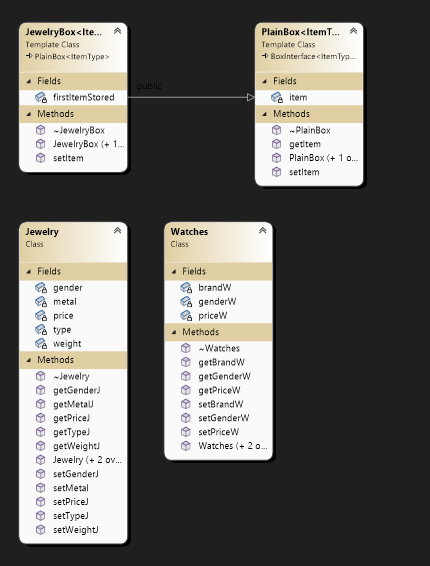


TABLA DESCRIPTIVA

JewelryBox.h

Tabla de atributos Private:

|  |  |
| --- | --- |
| bool firstItemStored; | Boolean that states if setItem() shall execute or not |

Tabla de atributos Public:

|  |  |
| --- | --- |
| JewelryBox(); | Constructur |
| JewelryBox(const ItemType&); | Copy Constructor |
| ~JewelryBox(); | Destructor |
| void setItem(const ItemType&); | Saves the value of ItemType |

BoxInterface.h

Tabla de atributos Public:

|  |  |
| --- | --- |
| ~BoxInterface() {} | Destructor |
| setItem(const ItemType& aItem) = 0; | Copy Constructor |
| getItem() const = 0; | Getter de data typpe “ItemType”. De setItem() |

PlainBox.h

Tabla de atributo Private:

|  |  |
| --- | --- |
| item | ItemTyoe variable |

Tabla de atributo Public:

|  |  |
| --- | --- |
| PlainBox(); | Constructor |
| PlainBox(const ItemType&) | Copy Constructor |
| ~PlainBox() | Destructor |
| setItem(const ItemType&) | Recevives variable of ItemType which is by inheretence. |
| getItem() const | Gets ItemType data type variables |

Jewelry.h

Tabla de atributos Private:

|  |  |
| --- | --- |
| genderJ | Pointer variable for gender |
| typeJ | Pointer variable for the type of jewelry |
| metalJ | Pointer variable for the type of metal |
| weightJ | Integer variable that receives the weight in carats |
| priceJ | Double variable for price |

Tabla de atributos Public:

|  |  |
| --- | --- |
| Jewelry(); | Constructor |
| Jewelry(char\*, char\*, int, double, char\*); | Parametrized constructor |
| Jewelry(const Jewelry&); | Copy constructor |
| ~Jewelry(); | Destructor |
| setGenderJ(char\*) | Receives the genderJ variable value |
| setTypeJ(char\*) | Receives the typeJ variable value |
| setMetal(char\*) | Receives the metalJ variable value |
| setWeightJ(int) | Receives the weightJ variable value |
| setPriceJ(double) | Receives the priceJ variable value |
| getGenderJ() const | Gets gender variable from setGenderJ method |
| getTypeJ() const | Gets jewelry’s type variable from setTypeJ method |
| getMetalJ() const | Gets metal variable from setMetalJ method |
| getWeightJ() const | Gets weight variable setWeightJ method |
| getPriceJ() const | Gets price variable setPriceJ method |
| ostream& operator<<(ostream&, const Jewelry&) | Jewels Output operator |
| istream& operator>>(istream&, Jewelry&) | Jewels Input operator |

Watch.h

Tabla de atributos Private:

|  |  |
| --- | --- |
| genderW | Variable tipo puntero |
| brandW | Variable tipo puntero |
| priceW | Variable tipo puntero |

Tabla de atributos Public:

|  |  |
| --- | --- |
| Watches() | Constructor |
| Watches(char\*, char\*, double) | Parametrized constructor |
| Watches(const Watches&) | Copy constructor |
| ~Watches() | Destructor |
| setGenderW(char\*) | Receives the genderW variable value |
| setBrandW(char\*) | Receives the brandW variable value |
| setPriceW(double) | Receives the priceW variable value |
| getGenderW() const | Gets genderW variable from setGenderW |
| getBrandW() const | Gets brand variable from setBrandW |
| getPriceW() const | Gets price variable setPriceW method |
| ostream& operator<<(ostream&, const Watches&); | Watches Output operator |
| istream& operator>>(istream&, Watches&); | Watches Input operator |

Main.cpp

#include "BoxInterface.h"

#include "JewelryBox.h"

#include "Jewelry.h"

#include "PlainBox.h"

#include "Watches.h"

#include <iostream>

#include <string>

using namespace std;

int main()

{

BoxInterface<Jewelry> \*joyas = new JewelryBox <Jewelry>;

BoxInterface<Watches> \*relojes = new JewelryBox <Watches>;

Jewelry jewels;

Watches watch;

cin >> jewels;

cout << endl;

cin >> watch;

cout << endl;

joyas->setItem(jewels);

relojes->setItem(watch);

cout << endl;

cout << joyas->getItem() << endl;

cout << relojes->getItem() << endl;

cout << endl;

delete joyas;

delete relojes;

system("pause");

return 0;

}

JewelryBox.h

#ifndef \_JEWELRY\_BOX

#define \_JEWELRY\_BOX

#include "PlainBox.h"

template <class ItemType>

class JewelryBox : public PlainBox<ItemType>

{

private:

bool firstItemStored;

public:

JewelryBox();

JewelryBox(const ItemType&);

~JewelryBox();

void setItem(const ItemType&);

};

template<class ItemType>

JewelryBox<ItemType>::JewelryBox() :PlainBox<ItemType>()

{

firstItemStored = false;

}

template<class ItemType>

JewelryBox<ItemType>::JewelryBox(const ItemType& aItem) : PlainBox<ItemType>()

{

PlainBox<ItemType> ::setItem(aItem);

firstItemStored = false;

}

template<class ItemType>

JewelryBox<ItemType>::~JewelryBox()

{

}

template<class ItemType>

void JewelryBox<ItemType>::setItem(const ItemType& aItem)

{

if (!firstItemStored)

{

PlainBox<ItemType> ::setItem(aItem);

firstItemStored = true;

}

}

#endif

Jewelry.h

#pragma once

#include<iostream>

#include<string>

using namespace std;

class Jewelry {

private:

char\* genderJ;

char\* typeJ;

char\* metalJ;

int weightJ;

double priceJ;

public:

Jewelry();// Default Constructor

Jewelry(char\*, char\*, int, double, char\*);// Constructor with parameters

Jewelry(const Jewelry&);// Copy Constructor

~Jewelry();// Destructor

// Mutators

void setGenderJ(char\*);

void setTypeJ(char\*);

void setMetalJ(char\*);

void setWeightJ(int);

void setPriceJ(double);

// Accessors

char\* getGenderJ() const;

char\* getTypeJ() const;

char\* getMetalJ() const;

int getWeightJ() const;

double getPriceJ() const;

friend ostream& operator<<(ostream&, const Jewelry&);

friend istream& operator>>(istream&, Jewelry&);

};

Watches.h

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Watches

{

private:

char\* genderW;

char\* brandW;

double priceW;

public:

// Default Constructor

Watches();

// Constructor with parameters

Watches(char\*, char\*, double);

// Copy Constructor

Watches(const Watches&);

// Destructor

~Watches();

// Mutators

void setGenderW(char\*);

void setBrandW(char\*);

void setPriceW(double);

// Accessors

char\* getGenderW() const;

char\* getBrandW() const;

double getPriceW() const;

friend ostream& operator<<(ostream&, const Watches&);

friend istream& operator>>(istream&, Watches&);

};

Jewelry.cpp

#include "Jewelry.h"

Jewelry::Jewelry()

{

char tempGTM[20] = " ";

setGenderJ(tempGTM);

setTypeJ(tempGTM);

setWeightJ(0);

setPriceJ(0);

setMetal(tempGTM);

}

Jewelry::Jewelry(char\* aGender, char\* aType, int aWeight, double aPrice, char\* aMetal)

{

setGenderJ(aGender);

setTypeJ(aType);

setWeightJ(aWeight);

setPriceJ(aPrice);

setMetal(aMetal);

}

Jewelry::Jewelry(const Jewelry& obj)

{

setGenderJ(obj.getGenderJ());

setTypeJ(obj.getTypeJ());

setWeightJ(obj.getWeightJ());

setPriceJ(obj.getPriceJ());

setMetal(obj.getMetalJ());

}

Jewelry::~Jewelry()

{

delete [] gender;

delete [] type;

delete [] metal;

}

void Jewelry::setGenderJ(char\* aGender)

{

gender = new char[strlen(aGender) + 1];

strcpy\_s(gender, strlen(aGender) + 1, aGender);

}

void Jewelry::setTypeJ(char\* aType)

{

type = new char[strlen(aType) + 1];

strcpy\_s(type, strlen(aType) + 1, aType);

}

void Jewelry::setWeightJ(int aWeight)

{

weight = aWeight;

}

void Jewelry::setPriceJ(double aPrice)

{

price = aPrice;

}

void Jewelry::setMetal(char\* aMetal)

{

metal = new char[strlen(aMetal) + 1];

strcpy\_s(metal, strlen(aMetal) + 1, aMetal);

}

char\* Jewelry::getGenderJ() const

{

return gender;

}

char\* Jewelry::getTypeJ() const

{

return type;

}

int Jewelry::getWeightJ() const

{

return weight;

}

double Jewelry::getPriceJ() const

{

return price;

}

char\* Jewelry::getMetalJ() const

{

return metal;

}

ostream& operator<<(ostream& strm, const Jewelry& obj)

{

strm << "La joya tiene las siguientes especificaciones:"

<< "\n Gender: " << obj.getGenderJ()

<< "\n Jewelry Type: " << obj.getTypeJ()

<< "\n Gold Metal Weight: " << obj.getWeightJ()

<< "\n Price: $" << obj.getPriceJ()

<< "\n Metal Type: " << obj.getMetalJ();

return strm;

}

istream& operator>>(istream& strm, Jewelry& obj)

{

cout << "Escoja el genero para quien sera la joya - MUJER, VARON, NINO: ";

strm >> obj.gender;

cout << "Escoja el tipo de joya - Opciones: ANILLOS, CADENAS, BRAZALETES, ARETES: ";

strm >> obj.type;

cout << "Especifique los KILATES de oro de la joya: ";

strm >> obj.weight;

cout << "Entre el precio: $";

strm >> obj.price;

cout << "Entre el tipo de metal que sera la joya: ";

strm >> obj.metal;

return strm;

}

Watches.cpp

#include "Watches.h"

Watches::Watches()

{

char temp[10] = " ";

setGenderW(temp);

setBrandW(temp);

setPriceW(0);

}

Watches::Watches(char\* genderW, char\* brandW, double priceW)

{

setGenderW(genderW);

setBrandW(brandW);

setPriceW(priceW);

}

Watches::Watches(const Watches& obj)

{

setGenderW(obj.getGenderW());

setBrandW(obj.getBrandW());

setPriceW(obj.getPriceW());

}

Watches::~Watches()

{

delete [] genderW;

delete [] brandW;

}

void Watches::setGenderW(char\* aGenderW)

{

genderW = new char[strlen(aGenderW) + 1];

strcpy\_s(genderW, strlen(aGenderW) + 1, aGenderW);

}

void Watches::setBrandW(char\* aBrandW)

{

brandW = new char[strlen(aBrandW) + 1];

strcpy\_s(brandW, strlen(aBrandW) + 1, aBrandW);

}

void Watches::setPriceW(double aPriceW)

{

priceW = aPriceW;

}

char\* Watches::getGenderW() const

{

return genderW;

}

char\* Watches::getBrandW() const

{

return brandW;

}

double Watches::getPriceW() const

{

return priceW;

}

ostream& operator<<(ostream& strm, const Watches& obj)

{

strm << "\nEl reloj que ha escojido tiene estas especificaciones:";

strm << "\n Genero: " << obj.getGenderW();

strm << "\n Brand: " << obj.getBrandW();

strm << "\n Precio: $" << obj.getPriceW() << endl;

return strm;

}

istream& operator>>(istream& strm, Watches& obj)

{

cout << "Escoja el genero para quien sera el reloj MUJER, VARON, NINO: ";

strm >> obj.genderW;

cout << "Especifique la marca del reloj que busca. Ejemplos Ferrary, COACH, Casio, Bilova, Citizen, Boos, etc. : ";

strm >> obj.brandW;

cout << "Por favor, entre precio: $";

strm >> obj.priceW;

return strm;

}

OUTPUT

Escoja el genero para quien sera la joya - MUJER, VARON, NINO: NINO

Escoja el tipo de joya - Opciones: ANILLOS, CADENAS, BRAZALETES, ARETES: ARETES

Especifique los KILATES de oro de la joya: 12

Entre el precio: $123

Entre el tipo de metal que sera la joya: ORO

Escoja el genero para quien sera el reloj MUJER, VARON, NINO: VARON

Especifique la marca del reloj que busca. Ejemplos Ferrary, COACH, Casio, Bilova, Citizen, Boos, etc. : COACH

Por favor, entre precio: $1223

La joya tiene las siguientes especificaciones:

Gender: NINO

Jewelry Type: ARETES

Gold Metal Weight: 12K

Price: $123

Metal Type: ORO

El reloj que ha escojido tiene estas especificaciones:

Genero: VARON

Brand: COACH

Precio: $1223

Press any key to continue . . .

PROMPT

