МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ

УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Лабораторна робота №1

з дисципліни: «Системне програмування»

СТВОРЕННЯ ДОДАТКУ НА МОВІ C++ В POSIX-СУМІСНІЙ СИСТЕМІ

Варіант 1

СПЕЦІАЛЬНОСТІ

121 – Інженерія програмного забезпечення

СПЕЦІАЛІЗАЦІЇ

Інженерія програмного забезпечення комп’ютерних систем

Виконали:

Балукова В. Кушніренко Ю. Курдій Д.

Підпис:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів:\_\_\_\_\_\_\_ Оцінка \_\_\_\_\_\_

Група: ІТ-93

Викладач: Катін П.Ю.

Підпис:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ 2021

**Завдання лабораторної роботи**

Інформаційна система **Бібліотеки**.

Клас

Працівники (Код співробітника, ПІБ, Вік, Пол, Адреса, Паспортні дані, Код посади) [10бзаписів].

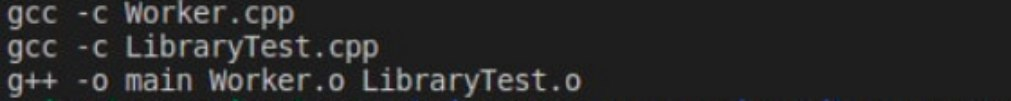
Передбачити функціональність додавання інформації, вивід інформації на консоль.

Передбачити функціональність поведінки класу, в залежності від значення певних членів класу.

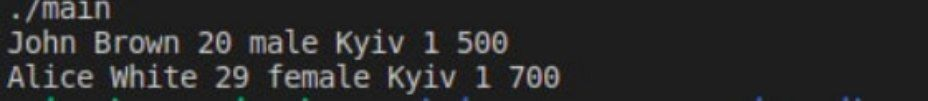
Під час тестування на етапі виконання створити екземпляри класу на купі пам'яті і на стеку.

**Етапи виконання лабораторної роботи**

Виконаємо команду make all:



Виконаємо команду make run:



Вихідний код програми:

Worker.cpp

#include <iostream>  
#include "Worker.h"  
  
using namespace std;  
  
Worker::Worker()  
{  
 // initialize data members  
 mPIB = "Unknown Worker";  
 mAge = 0;  
 mWorkerCode = 0;  
 mSex = "Unknown";  
 mPassport = "Unknown";  
 mPositionCode = 0;  
 mAddress = " Unknown";  
  
}  
  
Worker::~Worker()  
{  
 // nothing much to do in terms of cleanup  
}  
  
int Worker::calculateSalaryInDollars()  
{  
 if (getPositionCode() == 0) {  
  
 return 700;  
 }  
   
   
 return static\_cast<int>((getPositionCode() \* 500));  
}  
  
int Worker::getWorkerCode()  
{  
 return mWorkerCode;  
}  
  
void Worker::setWorkerCode(int inCode)  
{  
 mWorkerCode = inCode;  
}  
int Worker::getPositionCode()  
{  
 return mPositionCode;  
}  
  
void Worker::setPositionCode(int inCode)  
{  
 mPositionCode = inCode;  
}  
  
string Worker::getPIB()  
{  
 return mPIB;  
}  
  
void Worker::setPIB(string inPIB)  
{  
 mPIB = inPIB;  
}  
string Worker::getSex()  
{  
 return mSex;  
}  
  
void Worker::setSex(string inSex)  
{  
 mSex = inSex;  
}  
  
  
int Worker::getAge()  
{  
 return mAge;  
}  
  
void Worker::setAge(int inAge)  
{  
 mAge = inAge;  
}  
  
string Worker::getAddress()  
{  
 return mAddress;  
}  
  
void Worker::setAddress(string inAddress)  
{  
 mAddress = inAddress;  
}  
  
  
string Worker::getPassport()  
{  
 return mPassport;  
}  
  
void Worker::setPassport(string inPassport)  
{  
 mPassport = inPassport;  
}

Worker.h

#include <string>  
class Worker  
{  
 public:  
 Worker();  
 ~Worker();  
  
 int calculateSalaryInDollars();  
  
 int getWorkerCode();  
 void setWorkerCode(int code);  
 std::string getPIB();  
 void setPIB(std::string PIB);  
 std::string getSex();  
 void setSex(std::string sex);  
 int getAge();  
 void setAge(int age);  
 int getPositionCode();  
 void setPositionCode(int code);  
 std::string getAddress();  
 void setAddress(std::string address);  
 std::string getPassport();  
 void setPassport(std::string passport);  
 private:  
 int mWorkerCode;  
 std::string mPIB;  
 std::string mSex;  
 int mAge;  
 int mPositionCode;  
 std::string mAddress;  
 std::string mPassport;  
};

LibraryTest.cpp

#include <iostream>  
#include "Worker.h"  
  
using namespace std;  
  
int main(int argc, char\*\* argv)  
{  
 Worker WorkerStack;  
 Worker\* WorkerHeap;  
 WorkerHeap = new Worker();  
  
 WorkerStack.setAge(20);  
 WorkerStack.setPIB("John Brown");  
 WorkerStack.setSex("male");  
 WorkerStack.setAddress("Kyiv");  
 WorkerStack.setPositionCode(1);  
  
 WorkerHeap->setAge(29);  
 WorkerHeap->setPIB("Alice White");  
 WorkerHeap->setSex("female");  
 WorkerHeap->setAddress("Kyiv");  
  
 std::cout << WorkerStack.getPIB() << " "  
 << WorkerStack.getAge() << " "  
 << WorkerStack.getSex() << " "  
 << WorkerStack.getAddress() << " "  
 << WorkerStack.getPositionCode() << " "  
 << WorkerStack.calculateSalaryInDollars()  
 << std::endl;  
  
  
 std::cout << WorkerHeap->getPIB() << " "  
 << WorkerHeap->getAge() << " "  
 << WorkerHeap->getSex() << " "  
 << WorkerHeap->getAddress() << " "  
 << WorkerStack.getPositionCode() << " "  
 << WorkerHeap->calculateSalaryInDollars()   
 << std::endl;  
  
 delete WorkerHeap;  
  
 return 0;  
}

Makefile

all: helloworld.cpp  
 @gcc -c helloworld.cpp  
 @g++ -o main helloworld.o  
 @./main  
clean:  
 @rm -f \*.o   
 @rm -f \*.c  
 @rm main  
run:  
 @./main

**Висновок**

На цій лабораторній роботі ми навчилися створювати класи на C++ з розбиттям на файли декларацій та реалізацій. Логіка певних методів змінюється в залежності від значень полів класу.