lab 3

source

ex 0

```
#include <iostream>
#include <string>
using namespace std;

class Student {
public:
  string description = "A student of a group";
  void printDescription();
  Student();
};

Student::Student()
{
    cout << "Creating student object class named: " << description << endl;
}

class Chairman : private Student {
public:
  string description = "A chairman of a group";
  void printDescription();
};

void Student::printDescription() {
  cout << "Object description: " << description << endl;
}

void Chairman::printDescription()
{
    // Student::printDescription();
    cout << "Object description: " << description << endl;
}

int main() {
    Chairman chair;
    chair.printDescription();
}
```

```
output:
Creating student object class named: A student of a group
Object description: A chairman of a group
```

ex 1

- program doesn't compile because methods `informTeacher()` and `informGroup` are not accessible from `main()`
- methods are accessible from the `Coordinator` constructor

```
#include <iostream>
#include <string>
using namespace std;

class Student {
protected:
  string name_surname = "NO_NAME";
  unsigned int id_number = 0;

public:
  Student(string sname_surname, unsigned int sid_number);
  string description = "student of group";
  void printDescription();
  void printData(){
    cout << " Method printData of the Student class" << endl;
    cout << " name and surname " << name_surname << endl;
    cout << " id number " << id_number << endl;
  }
};

class Coordinator
{
    public:
        string description = "coordinator of a group";
    protected:
        Coordinator();
        void informGroup();
        void informTeacher();
};

Coordinator::Coordinator()
{
    informGroup();
    informTeacher();
    cout << "Creating an object of the Coordinator class named: " <<
description << std::endl;
}

void Coordinator::informGroup()
{
    std::cout<< "Group informed!!!\n";
}

void Coordinator::informTeacher()
{
    std::cout<< "Teacher informed!!!\n";
}

class Chairman : public Student, protected Coordinator{
private:
  string email = "no@noemail";
public:
  Chairman(string sname_surname, unsigned int sid_number, string semail);
  string description = "chairman of a group";
};

Chairman::Chairman(string sname_surname,
                   unsigned int sid_number,
                   string semail)
  : Student(sname_surname, sid_number)
  , Coordinator()
  , email(semail) {
  cout << "Creating an object of the Chairman class named: "
       << description << endl;
}

Student::Student(string sname_surname, unsigned int sid_number)
  : name_surname(sname_surname){
  id_number = sid_number;
  cout << "Creating an object of the Student class named: "
       << description << endl;
}

void Student::printDescription(){
  cout << "Description: " << description << endl;
}

int main(){
  Student stud("Jan Kowalski", 7);
  stud.printDescription();
  Chairman chair("Aleksandra Nowak", 999, "mail@nomail.dot");
  chair.printDescription();
//   chair.informGroup();   Coordinator methods are not accessible from
//   chair.informTeacher(): main function
}
```

```
output:
Creating an object of the Student class named: student of group
Description: student of group
Creating an object of the Student class named: student of group
Group informed!!!
Teacher informed!!!
Creating an object of the Coordinator class named: coordinator of a group
Creating an object of the Chairman class named: chairman of a group
Description: student of group
```

ex 2

replacing `printDescription()` declaration with `virtual void printDescription()=0` couses error:

> object of bastrack class type "Student" is not allowed: funciton "Student::printDescription" si a pure virtual function

object of `Student` type can not call mathod `printDescription()` because it is pure virtual method and doesn't have definition at compile time there for method has to be implemented by derived class

```
#include <iostream>
#include <string>
using namespace std;

class Student {
protected:
  string name_surname = "NO_NAME";
  unsigned int id_number = 0;

public:
  Student(string sname_surname, unsigned int sid_number);
  string description = "student of group";
  virtual void printDescription()=0;
  void printData(){
    cout << " Method printData of the Student class" << endl;
    cout << " name and surname " << name_surname << endl;
    cout << " id number " << id_number << endl;
  }
};

class Coordinator
{
    public:
        string description = "coordinator of a group";
    protected:
        Coordinator();
        void printDescription();
        void informGroup();
        void informTeacher();
};

Coordinator::Coordinator()
{
    // informGroup();
    // informTeacher();
    cout << "Creating an object of the Coordinator class named: " <<
description << std::endl;
}

void Coordinator::informGroup()
{
    std::cout<< "Group informed!!!\n";
}

void Coordinator::informTeacher()
{
    std::cout<< "Teacher informed!!!\n";
}

class Chairman : public Student, protected Coordinator{
private:
  string email = "no@noemail";
public:
  void printDescription();
  Chairman(string sname_surname, unsigned int sid_number, string semail);
  string description = "chairman of a group";
};

Chairman::Chairman(string sname_surname,
                   unsigned int sid_number,
                   string semail)
  : Student(sname_surname, sid_number)
  , Coordinator()
  , email(semail) {
  cout << "Creating an object of the Chairman class named: "
       << description << endl;
}

Student::Student(string sname_surname, unsigned int sid_number)
  : name_surname(sname_surname){
  id_number = sid_number;
  cout << "Creating an object of the Student class named: "
       << description << endl;
}

void Student::printDescription(){
  cout << "Description: " << description << endl;
}

void Coordinator::printDescription()
{
    cout << "Description: " << description << endl;
}

void Chairman::printDescription()
{
    cout << "Description: " << description << endl;
}

int main(){
  // Student stud("Jan Kowalski", 7);   ERROR! purely virtual method
  // stud.printDescription();
  Chairman chair("Aleksandra Nowak", 999, "mail@nomail.dot");
  chair.Student::printDescription();
  chair.Chairman::printDescription();
  chair.printDescription();
}
```

ex 3

1. no 😭
2. error couses `Device u`. to declare class with pure virtual methods you need to declare it via derived class
3. `id_` and `data_` are not declared even though they are used later in declaration of virtual methods in the same class

```
#include <iostream>
#include <sstream>
using namespace std;

class Device {
private:
  int id_;
  string data_;
public:
  virtual int write(int id, string data) = 0;
  virtual string read(int id) = 0;
};

class Disc : public Device {
private:
  int id_;
  string data_;
public:
  Disc(int id);
  int write(int id, string data);
  string read(int id);
};

Disc::Disc(int id){
  cout << "Creating an object of the Disc class   " << endl;
  id_ = id;
}

int Disc::write(int id, string data){
  if(id_ != id)
  {
    std::cout<<"identifiers not matching\n";
    return -1;
  }
  data_ = data;
  cout << "writing data: " << data << endl;
  return 1;
}
```

```
string Disc::read(int id){
  cout << "reading data: " << data_ << endl;
  return data_;
}

int main(){
//   Device u;
  Disc d1(7);
  d1.write(7, "test 11");
  d1.write(6, "test identifier");
  d1.read(7);
  system("pause");
}
```