Универзитет „Св. Кирил и Методиј“ - Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Аудиториски вежби 7

Повеќекратно наследување

Напреден развој на софтвер

Содржина

Задача 1

Да се напише класа Teacher која наследува особини од класите Employee и Person.

Задача 1

Решение 1/3

```
#include <iostream>
#include <cstring>
using namespace std;
class Person {
private:
    char name[50];
    int age;
    bool isMale;

public:
    Person(char *n, int a, bool m) {
        strcpy(name, n);
        age = a;
        isMale = m;
    }
    char * getName() {
        return name;
    }
    int getAge() {
        return age;
    }
    bool gender() {
        return isMale;
    }
};
```

Задача 1

Решение 2/3

```
class Employee {
private:
    char employer[50];
    double wage;

public:
    Employee(char *e, double w) {
        strcpy(employer, e);
        wage = w;
    }

    char* getEmployer() {
        return employer;
    }
    double getWage() {
        return wage;
    }
};
```

Задача 1
Решение 3/3

```
class Teacher: public Person, public Employee {
private:
    int numCourses;

public:
    Teacher(char *n, int a, bool m, char *e, double w, int c) :
        Person(n, a, m), Employee(e, w) {
        numCourses = c;
    }
    int getCourses() {
        return numCourses;
    }
    ;
};
int main() {
    Teacher t("Stojan Stojanov", 45, true, "Fakultet", 30000, 3);
    cout << "Nastavnik: " << t.getName() << "\nFakultet: " << t.getEmployer()
            << "\nPlata: " << t.getWage() << "\nBroj na predmeti: "
            << t.getCourses();
    return 0;
}
```

Задача 2

Да се состави класа за автомобил со млазен погон кој наследува својства од две класи, автомобил и млазен авион.

Задача 2

Решение 1/4

```
#include <iostream>
using namespace std;
class Vehicle {
public:
    Vehicle() {
        cout << "Vehicle Constructor" << endl;
    }
    virtual ~Vehicle() {
        cout << "Vehicle Destructor" << endl;
    }
    virtual void accelerate() {
        cout << "Vehicle Accelerating" << endl;
    }
    void setAcceleration(double a) {
        acceleration = a;
    }
    double getAcceleration() {
        return acceleration;
    }
protected:
    double acceleration;
};
```

Задача 2

Решение 2/4

```
class Car: public Vehicle {
public:
    Car() {
        cout << "Car Constructor" << endl;
    }
    virtual ~Car() {
        cout << "Car Destructor" << endl;
    }
    virtual void accelerate() {
        cout << "Car Accelerating" << endl;
    }
    virtual void drive() {
        cout << "Car Driving" << endl;
    }
};
class Jet: public Vehicle {
public:
    Jet() {
        cout << "Jet Constructor" << endl;
    }
    virtual ~Jet() {
        cout << "Jet Destructor" << endl;
    }
    virtual void fly() const {
        cout << "Jet flying" << endl;
    }
};
```

Задача 2

Решение 3/4

```
class JetCar: public Car, public Jet {
public:
    JetCar() {
        cout << "JetCar Constructor" << endl;
    }
    virtual ~JetCar() {
        cout << "JetCar Destructor" << endl;
    }
    virtual void drive() {
        cout << "JetCar driving" << endl;
    }
    virtual void fly() {
        cout << "JetCar flying" << endl;
    }
};
void analyzeCarPerformance(Car *testVehicle) {
    testVehicle->drive();
    //drive() exists for both base and sub class
}
void analyzeJetPerformance(Jet *testVehicle) {
    testVehicle->fly();
    //fly() exists for both base and sub class
}
```

Задача 2

Решение 4/4

```
int main() {
    Car myCar;
    Jet myJet;
    JetCar myJetCar;
    cout << endl << endl;
    cout << "Car testing in progress" << endl;
    analyzeCarPerformance(&myCar);
    analyzeCarPerformance(&myJetCar);
    cout << "Jet testing in progress" << endl;
    analyzeJetPerformance(&myJet);
    analyzeJetPerformance(&myJetCar);
    cout << endl << endl;
    return 0;
}
```

Материјали и прашања

Предавања, аудиториски вежби, соопштенија
courses.finki.ukim.mk

Изворен код на сите примери и задачи
bitbucket.org/tdelev/finki-nrs

Прашања и одговори
qa.finki.ukim.mk