Tampereen ammattikorkeakoulu

Sovellusohjelmoinnin jatkokurssi

Oppimispäiväkirja

Janne Lankinen

SISÄLLYS

1 Viikkotehtävät 3
1.1 Teht 1: 3
1.2 Teht 2: 5
1.3 Teht 3: 7
2 Viikkotehtävät 11
2.1 teht 1 11
2.2 Teht 2 14
2.3 Teht 3 17
2.4 Teht 4 20
3 Viikkotehtävät 24
4 Viikkotehtävät 25
5 Viikkotehtävät 26
6 Viikkotehtävät 27

1 Viikkotehtävät

1.1 Teht 1:

main.cpp:

```
#include <iostream>
#include <string>
#include "person.h"
#include "person.cpp"
using namespace std;

int main(){

    {
    Person Kalle;
    Kalle.setName("Kalle");
    Kalle.setAge(20);

    Person Ville;
    Ville.setName("Ville");
    Ville.setAge(23);

    Kalle.salute();
    Ville.salute();

    int x1 = Kalle.getAge();
    int x2 = Ville.getAge();

    cout << "Kalle is" << x1 << " years old." << endl;
    cout << "Ville is" << x2 << " years old." << endl;
    }

return 0;
}
```

person.cpp:

```
#include <string>

using namespace std;

class Person {
    private:
        string name;
        int age;
    public:
    void salute();
    void setAge(int newAge);
    int getAge();
    void setName(string newName);
    string getName();
};
```

person.h:

```
#include <string>

using namespace std;

class Person {
    private:
        string name;
        int age;
    public:
    void salute();
```

```
    void setAge(int newAge);
    int getAge();
    void setName(string newName);
    string getName();
};
```

1.2 Teht 2:

main.cpp:

```
#include <iostream>
#include <string>
#include "date.h"
#include "date.cpp"
using namespace std;

int main(){

    {
    Date date1;
    date1.setDate(1);
    date1.setMonth(1);
    date1.setYear(2020);

    Date date2;
    date2.setDate(2);
    date2.setMonth(2);
    date2.setYear(2020);

    date1.printDate();
    date2.printDate();
    }
    return 0;
}
```

date.cpp:

```
#include <iostream>
#include <string>
#include "date.h"
#include "date.cpp"
using namespace std;

int main(){

    {
    Date date1;
    date1.setDate(1);
    date1.setMonth(1);
    date1.setYear(2020);

    Date date2;
    date2.setDate(2);
    date2.setMonth(2);
    date2.setYear(2020);

    date1.printDate();
    date2.printDate();
    }
    return 0;
}
```

date.h:

```
#include <string>

using namespace std;

class Date {
    private:
```

```
    int date;
    int month;
    int year;

    public:
  void setDate(int newDate);
  void setMonth(int newMonth);
  void setYear(int newYear);
  int getDate();
  int getmMonth();
  int getYear();
  void printDate(string format);
  void printDate();
};
```

1.3 Teht 3:

main.cpp:

```
#include <iostream>
#include <string>
#include "date.h"
#include "date.cpp"
using namespace std;

int main() {
  Date date1;
  date1.askDate();
  date1.printDate();

  date1.addOneDay();
  date1.printDate();

  return 0;
}
```

date.cpp:

```
#include "date.h"
#include <iostream>

using namespace std;

class Date {
public:
    void setDate(int newDate);
    int getDate();
    void setMonth(int newMonth);
    int getMonth();
    void setYear(int newYear);
    int getYear();
    void printDate();
    void printDate(string format);
    void askDate();
    void addOneDay();
private:
    int date;
    int month;
    int year;
};

void Date::setDate(int newDate) {
    date = newDate;
}

int Date::getDate() {
    return date;
}

void Date::setMonth(int newMonth) {
    month = newMonth;
```

```
}

int Date::getMonth() {
    return month;
}

void Date::setYear(int newYear) {
    year = newYear;
}

int Date::getYear() {
    return year;
}

void Date::printDate() {
    cout << date << "/" << month << "/" << year << endl;
}

void Date::printDate(string format) {
    // Implement custom format printing if needed
}

void Date::askDate() {
    cout << "Enter day: ";
    cin >> date;
    cout << "Enter month: ";
    cin >> month;
    cout << "Enter year: ";
    cin >> year;
}

void Date::addOneDay() {
    date++;
    if (date > 30) { // Simplified month length handling
        date = 1;
```

```
        month++;
        if (month > 12) {
            month = 1;
            year++;
        }
    }
}
```

date.h:

```
#include <string>

using namespace std;

class Date {
    private:
        int date;
        int month;
        int year;

        public:
    void setDate(int newDate);
    void setMonth(int newMonth);
    void setYear(int newYear);
    int getDate();
    int getmMonth();
    int getYear();
    void printDate(string format);
    void printDate();
    void askDate();
    void addOneDay();
};
```

2 Viikkotehtävät

2.1 teht 1

main.cpp:

```
#include <iostream>
#include <string>
#include <cstdlib>
#include <ctime>
#include "noppa.h"

using namespace std;

int main () {

    setlocale(LC_ALL, "fi_FI");

    int gameChoice;
    cout << "Choose a game: 1. Monopoly 2. Yatzy" << endl;
    cin >> gameChoice;

    if (gameChoice == 1) {
        Cube monopoly(2);
        monopoly.throwCube();
        monopoly.showLatestThrow();
    }
    else if (gameChoice == 2) {
        Cube yatzy(5);
        yatzy.throwCube();
        yatzy.showLatestThrow();
        std::cout << "results";
    }

    srand(time(0));

    Cube Noppa;
```

```
    Noppa.throwCube();
    Noppa.showLatestThrow();

    return 0;
}
```

person.cpp:

```
#include "person.h"

Person::Person() {
    name = "Unknown";
    age = 0;
}

Person::Person(std::string name, int age) {
    this->name = name;
    this->age = age;
}

Person::~Person() {
    std::cout << "Person " << name << " is being destroyed." << std::endl;
}

void Person::setName(std::string name) {
    this->name = name;
}

void Person::setAge(int age) {
    this->age = age;
}

std::string Person::getName() {
    return name;
```

```
}

int Person::getAge() {
    return age;
}

void Person::salute() {
    std::cout << "Hello, my name is " << name << "!" << std::endl;
}

void Person::printPersonDetails() {
    std::cout << "Name: " << name << ", Age: " << age << std::endl;
}
```

person.h:

```
#ifndef PERSON_H
#define PERSON_H
#include <string>
#include <iostream>
using namespace std;

class Person {
    private:
        string name;
        int age;
    public:
    Person();
    Person(string n, int a);
    ~Person();

    void printPersonDetails();
    void salute();
    void setAge(int newAge);
```

```
    int getAge();
    void setName(string newName);
    string getName();
};

#endif
```

2.2 Teht 2

main.cpp:

```
#include <iostream>
#include <string>
#include "person.h"
using namespace std;

void createPerson() {
    Person tempPerson("Temporary", 30);
    tempPerson.printPersonDetails();
}

int main(){

    setlocale(LC_ALL,"fi_FI");

    cout << "Creating person at the beginning of main" << endl;
    Person Kalle;
    Kalle.setName("Kalle");
    Kalle.setAge(20);

    cout << "Creating person inside if block" << endl;
    if (true) {
        Person Ville;
        Ville.setName("Ville");
        Ville.setAge(23);
```

```
    Ville.salute();
}

cout << "Creating person inside for loop" << endl;
for (int i = 0; i < 2; i++){
    Person tempPerson("Jalmari", i + 20);
    tempPerson.printPersonDetails();
}

cout << "Creating person inside a function" << endl;
createPerson();

cout << "Creating dynamic person" << endl;
Person* pekka = new Person("Pekka", 20);
pekka->printPersonDetails();
delete pekka;

cout << "End of main" << endl;

return 0;
}
```

person.cpp:

```
#include "person.h"

Person::Person() {
    cout << "Person class default constructor" << endl;
    name = "";
    age = 0;
}

Person::Person(string name, int age) {
    cout << "Person class parameterized constructor" << endl;
    this->name = name;
```

```
    this->age = age;
}

Person::~Person() {
    cout << "Person class destructor for " << name << endl;
}

void Person::setName(string name) {
    this->name = name;
}

void Person::setAge(int age) {
    this->age = age;
}

string Person::getName() {
    return name;
}

int Person::getAge() {
    return age;
}

void Person::salute() {
    cout << "Hello, my name is " << name << " and I am " << age << " years old."
<< endl;
}

void Person::printPersonDetails() {
    cout << "Name: " << name << ", Age: " << age << endl;
}
```

person.h:

```
#ifndef PERSON_H
#define PERSON_H

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

class Person {
public:
    Person();
    Person(string name, int age);
    ~Person();
    void setName(string name);
    void setAge(int age);
    string getName();
    int getAge();
    void salute();
    void printPersonDetails();

private:
    string name;
    int age;
};

#endif
```

2.3 Teht 3

main.cpp:

```
#include <iostream>
#include <string>
#include <cstdlib>
```

```
#include <ctime>
#include "noppa.h"

using namespace std;

int main () {

    setlocale(LC_ALL, "fi_FI");

    srand(time(0));

    Cube Noppa;

    int latestThrow = Noppa.throwCube();
    Noppa.showLatestThrow();

    return 0;
}
```

noppa.cpp:

```
#include "noppa.h"

Cube::Cube() {
    latestThrow = 0;
}

Cube::~Cube() {
    std::cout << "cube object destroyed" << std::endl;
}

int Cube::throwCube() {
    latestThrow = rand() % 6 + 1;
    return latestThrow;
```

```
}

void Cube::showLatestThrow() {
    std::cout << "Latest throw: " << latestThrow << std::endl;
}
```

noppa.h:

```
#ifndef NOPPA_h
#define NOPPA_h
#include <iostream>
#include <string>
#include <cstdlib>
#include <ctime>
using namespace std;

class Cube {
    private:
        int latestThrow;
    public:
        Cube();
        ~Cube();

        int throwCube();
        void showLatestThrow();
};

#endif
```

2.4 Teht 4

main.cpp:

```
#include <iostream>
#include <string>
#include <cstdlib>
#include <ctime>
#include "noppa.h"

using namespace std;

int main () {

    setlocale(LC_ALL, "fi_FI");

    int gameChoice;
    cout << "Choose a game: 1. Monopoly 2. Yatzy" << endl;
    cin >> gameChoice;

    if (gameChoice == 1) {
        Cube monopoly(2);
        monopoly.throwCube();
        monopoly.showLatestThrow();
    }
    else if (gameChoice == 2) {
        Cube yatzy(5);
        yatzy.throwCube();
        yatzy.showLatestThrow();
```

```
        std::cout << "results";
    }

    srand(time(0));

    Cube Noppa;

    Noppa.throwCube();
    Noppa.showLatestThrow();

    return 0;
}
```

noppa.cpp:

```
#include "noppa.h"
#include <iostream>
#include <string>
#include <ctime>

Cube::Cube() : numDice(1) {
    std::cout << "cube class default constructuor" << std::endl;
    srand(time(0));
}

Cube::Cube(int numDice) : numDice(numDice) {
    std::cout << "cube parameterized constructor" << std::endl;
    srand(time(0));
}

Cube::~Cube () {}

void Cube::setNumDice(int numDice) {
    if (numDice >= 1 && numDice <= 5) {
        this->numDice = numDice;
```

```
    }
    else {
        std::cout << "Invalid number of dice. Number must be between 1-5" <<
std::endl;
    }
}

int Cube::getNumDice() {
    return numDice;
}

void Cube::throwCube() {
    latestThrows.clear();
    for (int i = 0; i < numDice; ++i) {
        latestThrows.push_back(rand() % 6 + 1);
    }
}

void Cube::showLatestThrow() {
    int sum = 0;
    for (int i = 0; i < latestThrows.size(); ++i) {
        cout << "Dice " << i + 1 << ": " << latestThrows[i] << endl;
        sum += latestThrows[i];
    }
    cout << "Total: " << sum << ". Thrown with " << numDice << " dice." << endl;
}
```

noppa.h:

```
#ifndef NOPPA_H
#define NOPPA_H

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

```
class Cube {
private:
    int numDice;
    vector<int> latestThrows;

public:
    Cube(); // Default constructor
    Cube(int numDice); // Parameterized constructor
    ~Cube();

    void setNumDice(int numDice);
    int getNumDice();
    void throwCube();
    void showLatestThrow();
};

#endif
```

3 Viikkotehtävät

4 Viikkotehtävät

5 Viikkotehtävät

6 Viikkotehtävät