Отчет по объектно-ориентированному программированию

студента группы КТ-31-24

Киселев М.А.

Вариант 14

Лабораторная работа 2.

**КЛАССЫ И ОБЪЕКТЫ В С++**

1. Задание

Создать класс Осадки с полями:

тип осадков - char \*

количество дней в году – int

время года – char \*

Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.

Определить в классе деструктор.

Определить в классе компоненты-функции для просмотра и установки полей данных.

Определить указатель на компоненту-функцию.

Определить указатель на экземпляр класса.

Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал).

Показать в программе использование указателя на объект и указателя на компоненту-функцию.

1. Код

precipitation.hpp:

#ifndef PRECIPITATION\_H

#define PRECIPITATION\_H

#include <string>

class Precipitation {

public:

/\* Constructors \*/

Precipitation();

Precipitation(const std::string \_sediment,  
 const int \_numberDaysInYear,   
 const std::string \_season);  
Precipitation(const Precipitation& \_prec);  
  
/\* Destructor \*/  
~Precipitation();  
  
/\* Getters \*/  
std::string getSediment();  
int getNumberdaysInYear();  
std::string getSeason();  
  
/\* Setters \*/  
void setSediment(std::string \_sediment);  
void setNumberDaysInYear(int \_numberDaysInYear);  
void setSeason(std::string \_season);  
  
/\* Additional functions \*/  
void printInf();

private:

std::string sediment;

int numberDaysInYear;

std::string season;

};

#endif

precipitation.cpp:

#include "../include/precipitation.hpp"

#include <string>

#include <iostream>

/\* Constructors \*/

Precipitation::Precipitation() { std::cout<< "Default constructor!" << std::endl; }

Precipitation::Precipitation(const std::string \_sediment, const int \_numberDaysInYear, const std::string \_season) {

sediment = \_sediment;

numberDaysInYear = \_numberDaysInYear;

season = \_season;

std::cout << "Constructor with arguments!" << std::endl;

}

Precipitation::Precipitation(const Precipitation& \_prec) {

sediment = \_prec.sediment;

numberDaysInYear = \_prec.numberDaysInYear;

season = \_prec.season;

std::cout << "Copy constructor" << std::endl;

}

/\* Destructor \*/

Precipitation::~Precipitation() { std::cout << "Destructor!" << std::endl; }

/\* Getters \*/

std::string Precipitation::getSediment() { return sediment; }

int Precipitation::getNumberdaysInYear() { return numberDaysInYear; }

std::string Precipitation::getSeason() { return season; }

/\* Setters \*/

void Precipitation::setSediment(std::string \_sediment) { sediment = \_sediment; }

void Precipitation::setNumberDaysInYear(int \_numberDaysInYear) { numberDaysInYear = \_numberDaysInYear; }

void Precipitation::setSeason(std::string \_season) { season = \_season; }

/\* Additional functions \*/

void Precipitation::printInf() {

std::cout << "Inf of Precipitation: " << std::endl;

std::cout << "\tsediment: " << sediment << std::endl;

std::cout << "\tnumberDaysInYear: " << numberDaysInYear << std::endl;

std::cout << "\tseason: " << season << std::endl;

}

main.cpp:

#include "./include/precipitation.hpp"

#include <cstdlib>

#include <ctime>

#include <iostream>

void generateData(Precipitation \*\_arr, const size\_t \_count);

Precipitation \*createArrPrecipitation(const size\_t \_count);

void deleteArrPrecipitation(Precipitation \*\_arr, const size\_t \_count);

void printArrPrecipitation(Precipitation \*const \_arr, int \_count);

int main() {

srand(time({}));

const int count = 10;  
Precipitation \*arrPrecipitation = createArrPrecipitation(count);  
printArrPrecipitation(arrPrecipitation, count);

deleteArrPrecipitation(arrPrecipitation, count);  
return 0;

}

/\* Generate arr Precipitation \*/

void generateData(Precipitation \*\_arr, const size\_t \_count) {

const std::string sedimentTypes[5] = { "rain", "snow", "hail", "sun", "acid rain" };

const std::string seasonTypes[4] = { "winter", "spring", "summer", "autumn" };

for (int i = 0; i < \_count; ++i) {  
 \_arr[i].setSediment(sedimentTypes[rand() % 5]);  
 \_arr[i].setNumberDaysInYear(rand() % 90);  
 \_arr[i].setSeason(seasonTypes[rand() % 4]);  
}

}

Precipitation \*createArrPrecipitation(const size\_t \_count) {

Precipitation \*arr = new Precipitation[\_count];

generateData(arr, \_count); return arr;

}

void deleteArrPrecipitation(Precipitation \*\_arr, const size\_t \_count) {

delete[](\_arr);

}

/\* Print arr Precipitation \*/ void printArrPrecipitation(Precipitation \*const \_arr, int \_count) { void (Precipitation::\*ptrPrintInf)() = &Precipitation::printInf;

for (int i = 0; i < \_count; ++i) {  
 std::cout << "Num: " << i+1 << std::endl;  
 (\_arr[i].\*ptrPrintInf)();  
 std::cout << std::endl;  
}

}

CMakeLists.txt:

cmake\_minimum\_required(VERSION 3.11)

project(Precipitation)

add\_executable(${PROJECT\_NAME} ./main.cpp ./source/precipitation.cpp)

target\_include\_directories(${PROJECT\_NAME} PRIVATE include)

1. Вывод

