Министерство образования Республики Беларусь
Учреждение образования
“Брестский Государственный технический университет”
Кафедра ИИТ

Лабораторная работа №3

По дисциплине “Проектирование программного обеспечения интеллектуальных систем”

Тема: “Ссылочный тип. Инициализация классов, конструкторы и деструкторы”

Выполнил:
Студент 2 курса
Группы ИИ-21
Кирилович А. А.
Проверил:
Монтик Н. С.

Брест 2022

Цель работы: Изучить использование ссылочного типа в пользовательских классах, запрограммировать классы с использованием конструкторов (по умолчанию, с параметрами, конструктор копирования) и деструкторов.

Ход работы: Разработать три пользовательских класса в соответствии с целью.

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

class LAB {
private:
	std::string name_;
	int number_;
	int mark_;
	std::string &name = name_;
	int &number = number_;
	int &mark = mark_;
public:
	LAB() {
		name = "Kirilovich";
		number = 1;
		mark = 4;
	}
	LAB(std::string &name, int &number, int &mark) {
		this->name = name;
		this->number = number;
		this->mark = mark;
	}
	LAB(const LAB &lab) {
		this->name = lab.name;
		this->number = lab.number;
		this->mark = lab.mark;
	}
	~LAB() {
		std::cout << “Destructor” << std::endl;
	}
	void set_object(std::string &name, int &number, int &mark) {
		this->name = name;
		this->number = number;
		this->mark = mark;
	}
	std::string get_name() {
		return name;
	}
	int& get_number() {
		return number;
	}
	int& get_mark() {
		return mark;
	}
	void print() {
		std::cout << "Name: " << name << std::endl;
		std::cout << "Number: " << number << std::endl;
		std::cout << "Mark: " << mark << std::endl;
	}
	LAB operator=(const LAB &lab) {
		this->name = lab.name;
		this->number = lab.number;
		this->mark = lab.mark;
		return *this;
	}
};

class Students_LAB {
private:
	LAB *lab;
	int size_;
	int &size = size_;
public:
	Students_LAB() {
		size = 1;
		lab = new LAB[size];
	}
	Students_LAB(int &size, LAB *lab) {
		this->size = size;
		this->lab = new LAB[size];
```

```
		for (int i = 0; i < size; i++) {
			this->lab[i] = lab[i];
		}
	}
	Students_LAB(const Students_LAB &students_lab) {
		this->size = students_lab.size;
		this->lab = new LAB[size];
		for (int i = 0; i < size; i++) {
			this->lab[i] = students_lab.lab[i];
		}
	}
	~Students_LAB() {
		delete[] lab;
	}
	void set_object(int &size, LAB *lab) {
		this->size = size;
		this->lab = new LAB[size];
		for (int i = 0; i < size; i++) {
			this->lab[i] = lab[i];
		}
	}
	int& get_size() {
		return size;
	}
	LAB& get_lab(int i) {
		return lab[i];
	}
	void print() {
		for (int i = 0; i < size; i++) {
			lab[i].print();
		}
	}
	Students_LAB operator=(const Students_LAB &students_lab) {
		this->size = students_lab.size;
		this->lab = new LAB[size];
		for (int i = 0; i < size; i++) {
			this->lab[i] = students_lab.lab[i];
		}
		return *this;
	}
};
class Group {
private:
	int count_;
	int &count = count_;
	std::string *students;
public:
	Group() {
		count = 1;
		students = new std::string[count];
		students[0] = "Ars";
	}
	Group(int &count, std::string *students) {
		this->count = count;
		this->students = new std::string[count];
		for (int i = 0; i < count; i++) {
			this->students[i] = students[i];
		}
	}
	Group(const Group &group) {
		this->count = group.count;
		this->students = new std::string[count];
		for (int i = 0; i < count; i++) {
			this->students[i] = group.students[i];
		}
	}
	~Group() {
		delete[] students;
	}
	void set_object(int &count, std::string *students) {
		this->count = count;
		this->students = new std::string[count];
		for (int i = 0; i < count; i++) {
			this->students[i] = students[i];
		}
	}
	int& get_count() {
```

```
		return count;
	}
	std::string& get_student(int i) {
		return students[i];
	}
	void print() {
		for (int i = 0; i < count; i++) {
			std::cout << students[i] << std::endl;
		}
	}
	Group operator=(const Group &group) {
		this->count = group.count;
		this->students = new std::string[count];
		for (int i = 0; i < count; i++) {
			this->students[i] = group.students[i];
		}
		return *this;
	}
};
int main() {
	std::string name = "Ivanov";
	int number = 2;
	int mark = 5;
	LAB lab1;
	lab1.print();
	std::cout << std::endl;
	LAB lab2(name, number, mark);
	lab2.print();
	std::cout << std::endl;
	LAB lab3(lab2);
	lab3.print();
	std::cout << std::endl;
	lab3.set_object(name, number, mark);
	lab3.print();
	std::cout << std::endl;
	lab3 = lab2;
	lab3.print();
	std::cout << std::endl;
	std::cout << lab3.get_name() << std::endl;
	std::cout << lab3.get_number() << std::endl;
	std::cout << lab3.get_mark() << std::endl;
	std::cout << std::endl;
	lab3.edit_ref(10);
	int size = 2;
	LAB *lab = new LAB[size];
	lab[0] = lab1;
	lab[1] = lab2;
	Students_LAB students_lab1;
	students_lab1.print();
	std::cout << std::endl;
	Students_LAB students_lab2(size, lab);
	students_lab2.print();
	std::cout << std::endl;
	Students_LAB students_lab3(students_lab2);
	students_lab3.print();
	std::cout << std::endl;
	students_lab3.set_object(size, lab);
	students_lab3.print();
	std::cout << std::endl;
	students_lab3 = students_lab2;
	students_lab3.print();
	std::cout << std::endl;
	std::cout << students_lab3.get_size() << std::endl;
	students_lab3.get_lab(0).print();
	students_lab3.get_lab(1).print();
	std::cout << std::endl;
	delete[] lab;
	Group group1;
	group1.print();
	std::cout << std::endl;
	int count = 2;
	std::string students[2] = {"Petrov", "Sidorov"};
	Group group2(count, students);
	group2.print();
	std::cout << std::endl;
	Group group3(group2);
	group3.print();
	std::cout << std::endl;
	group3.set_object(count, students);
	group3.print();
	std::cout << std::endl;
	std::cout << group3.get_count() << std::endl;
	std::cout << group3.get_student(0) << std::endl;
	std::cout << std::endl;
	group3 = group2;
	group3.print();
	std::cout << std::endl;

	}
```

Вывод: изучил использование ссылочного типав пользовательских классах, запрограммировал(по умолчанию, с параметрами,классы с использованием конструкторовконструктор копирования) и деструкторов.