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

**Лабораторная работа №4**

По дисциплине “Проектирование программного обеспечения интеллектуальных систем”

Тема: “Функции-друзья классов. Перегрузка операторов классов”

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

Брест 2022

**Цель работы:** изучение использования friend-функций для доступа к классам извне. Изучение особенностей перегрузки операторов в пользовательских классах.

**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_;
	friend void set_object1(LAB &lab, std::string &name, int &number, int &mark);
	friend std::string get_name1( LAB &lab);
	friend int& get_number1(LAB &lab);
	friend int& get_mark1(LAB &lab);
	friend void edit_ref1(LAB &lab, int num);
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 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;
	}
};

void set_object1(LAB &lab, std::string &name, int &number, int &mark) {
	lab.name = name;
	lab.number = number;
	lab.mark = mark;
}
std::string get_name1( LAB &lab) {
	return lab.name;
}
int& get_number1( LAB &lab) {
	return lab.number;
}
int& get_mark1(LAB &lab) {
	return lab.mark;
}
void edit_ref1(LAB &lab, int num) {
	std::cout << "Edit ref" << std::endl;
	std::cout << "Number: " << lab.number << std::endl;
	lab.number = num;
	std::cout << "Number: " << lab.number << std::endl;
}

class Group {
private:
```

```
	int count_;
	int &count = count_;
	std::string *students;
	friend void set_object2(Group &group , int &count, std::string *students);
	friend int& get_count2(Group &group);
	friend std::string& get_student2(Group &group, int i);
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 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;
	}
};

void set_object2(Group &group , int &count, std::string *students) {
	group.count = count;
	group.students = new std::string[count];
	for (int i = 0; i < count; i++) {
		group.students[i] = students[i];
	}
}
int& get_count2( Group &group) {
	return group.count;
}
std::string& get_student2( Group &group, int I) {
	return group.students[i];
}

class Students_LAB {
private:
	LAB *lab;
	int size_;
	int &size = size_;
	friend void set_object3(Students_LAB &students_lab, int &size, LAB *lab);
	friend int& get_size3( Students_LAB &students_lab);
	friend LAB& get_lab3( Students_LAB &students_lab, int i);
	friend void print3(Students_LAB &students_lab);
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;
	}
	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;
	}
};

void set_object3(Students_LAB &students_lab, int &size, LAB *lab) {
	students_lab.size = size;
	students_lab.lab = new LAB[size];
	for (int i = 0; i < size; i++) {
		students_lab.lab[i] = lab[i];
	}
}
int& get_size3( Students_LAB &students_lab) {
	return students_lab.size;
}
LAB& get_lab3( Students_LAB &students_lab, int I) {
	return students_lab.lab[i];
}
void print3(Students_LAB &students_lab) {
	for (int i = 0; i < students_lab.size; i++) {
		students_lab.lab[i].print();
	}
}

int main() {
	std::string name = "Lab1";
	int number = 1;
	int mark = 5;
	LAB lab;
	set_object1(lab, name, number, mark);
	std::cout << "Name: " << get_name1(lab) << std::endl;
	std::cout << "Number: " << get_number1(lab) << std::endl;
	std::cout << "Mark: " << get_mark1(lab) << std::endl;
	edit_ref1(lab, 2);
	std::cout << "Number: " << get_number1(lab) << std::endl;
	std::cout << "---------------------------------" << std::endl;
	int count = 3;
	std::string students[3] = { "Ars", "Vlad", "Vova" };
	Group group;
	set_object2(group, count, students);
	group.print();
	std::cout << "Count: " << get_count2(group) << std::endl;
	std::cout << "Student: " << get_student2(group, 1) << std::endl;
	std::cout << "---------------------------------" << std::endl;
	int size = 2;
	LAB lab1;
	set_object1(lab1, name, number, mark);
	LAB lab2;
	set_object1(lab2, name, number, mark);
	LAB lab3;
	set_object1(lab3, name, number, mark);
	LAB lab4;
	set_object1(lab4, name, number, mark);
	LAB lab5;
	set_object1(lab5, name, number, mark);
	LAB lab6;
	set_object1(lab6, name, number, mark);
	LAB lab7;
}
```