

Пример варианта заданий на перезачёте

1. По схеме составить иерархию классов.



2. Объяснить, что такое инкапсуляция, добавить сеттеры и геттеры к написанной структуре классов, объяснить их назначение.
3. Объяснить, что такое полиморфизм. Добавить к классу человек таблицу планов 7 на 24. Реализовать проверку занятости человека в заданный день, в заданный час (минуты не учитываем). Переопределить данную функцию для учителя.
4. Задание совпадает с предыдущим, однако переопределить функцию необходимо для студента.

Замечания к заданиям: для 3 и 4 заданий вы можете ввести какие-то упрощения, которые относятся к конкретной программной реализации и мало связаны с ООП. Пример упрощений в решении приведён.

Решение к №1:

```
class Human {
    std::string name;
    std::string birthday;
    int gender;
};

class Course {
    int id;
    std::string specialization;
};

class TimeTable {
    Course table[6][7];
    int fullness;
};

class Student : public Human {
    int group_id;
    int recordBook;
    double score;
};

class Group {
    int id;
    Student* students;
```

```

    TimeTable schedule;
};

class Teacher : public Human {
    TimeTable schedule;
    std::string specialization;
    int hours;
};

```

Частичное решение к №2:

```

class Human {
    std::string name;
    std::string birthday;
    int gender;
public:
    std::string getName() {
        return name;
    };
    void setName(std::string _name) {
        name = _name;
    };
};

```

Частичное решение к №3:

Пояснения: для упрощения изложения ответа поле table в классе TimeTable считалось публичным.

Функция для человека:

```

virtual bool checkFree(int day, int time) {
    if ((day > 0 && day < 8) && (time > 0 && time < 25)) {
        if (plans[day - 1][time - 1] != "") {
            std::cout << "You already have plan in this time: "
                << plans[day - 1][time - 1];
            return false;
        } else {
            return true;
        }
    } else {
        throw "error";
    }
};

```

Функция для учителя:

Пояснения: для упрощения ответа на вопрос функция convertor, переводящая время в номер пары, считается известной, также как и функция проверки ячейки расписания на пустоту.

```

virtual bool checkFree(int day, int time) {
    if ((day > 0 && day < 8) && (time > 0 && time < 25)) {
        if (Human::checkFree(day, time)) {
            if (day == 7)
                return true;
            int numOfLesson = convertor(time);
            if (!schedule.table[day - 1][numOfLesson - 1].isEmpty()) {
                std::cout << "You already have lesson in this time";
                return false;
            } else {

```

```
        return true;
    }
    } else {
        return false;
    }
} else {
    throw "error";
}
};
```

