Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа  
“Классы”. №1**

Выполнил:   
студент группы ИВТ-23-2Б   
Чудинов Данил Николаевич

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

Пермь, 2024 г.

**«Классы и объекты. Инкапсуляция»**

**Анализ задачи:**

Анализ задачи:

1. В задании необходимо реализовать класс `ArifmProgress`, у которого есть два приватных поля `first` (первый элемент прогрессии a₀) и `second` (постоянное отношение r).

2. У класса должны быть публичные методы `SetFirst()` для установки значения первого элемента и `SetSecond()` для установки значения постоянного отношения.

3. Требуется реализовать метод `element(int j)`, который вычисляет j-й элемент геометрической прогрессии по формуле aⱼ= a₀ \* rᐟⱿ.

4. Для возведения числа в степень используется приватный метод `poww`, который вызывается внутри метода `element()`. В данном случае, чтобы вычислить a ⱼ, мы умножаем первый элемент a₀ на r в степени j.

5. В функции `main()` создается объект класса `ArifmProgress`, устанавливаются значения первого элемента и постоянного отношения, после чего вызывается метод `element()` с аргументом 2 и выводится результат.

6. Программа выводит на экран все промежуточные шаги расчета.

7. На выходе программа выводит результат вычисления j-го элемента геометрической прогрессии по функции `element()` с точностью до 15 знаков после запятой.

Данный код верно реализует задачу и выполняет все требования указанные в задании.

**Код на языке C++:**

#include <iostream>

#include <cmath>

using namespace std;

class ArifmProgress {

private:

double first;

double second;

double poww(double x, double y) {

cout << "Возводим " << x << " в степень: " << y << endl;

double c = pow(x, y);

return c;

}

public:

void SetFirst(double x) {

this->first = x;

cout << "a0: " << first << endl;

}

void SetSecond(double y) {

this->second = y;

cout << "r: " << second << endl;

}

double element(double j) {

cout << "a0 ( " << first << " )" << " мы умножаем на следующее действие: " << endl;

return first \* poww(second, j);

}

};

int main() {

setlocale(LC\_ALL, "Rus");

ArifmProgress prog;

prog.SetFirst(4);

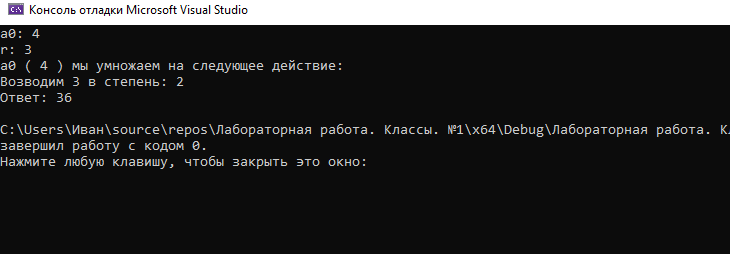
prog.SetSecond(3);

cout << "Ответ: " << prog.element(2) << endl;

return 0;

}

**Работа программы:**



**Ответ на вопросы:**

1) **Класс** – это абстрактный тип данных, определяемый пользователем. Представляет собой модель реального объекта в виде данных и функций для работы с ними.

2)Переменная класса называется экземпляром класса или объектом.

3)Переменными(Например: Class Auto: public: Lada(это объект) 2015 20000(это поля)

4) Методы

5) Чтобы ограничить работу с разными данными

6) Само его название говорит за себя, он находится в общем доступе(с ним можно работать из любой точки программы)

7) По умолчанию спецификатор доступа для всех полей и методов – private. С данными можно работать только внутри класса, чтобы скрыть их от других.

8) private

9) public

10) Я думаю, что с private. Ведь если разработчиков много, то работа с программой будет труднее.

11) Напрямую или же через функцию Seter

12) Напрямую или же через функцию Geter

13) S->name;

14) #include <iostream>

#include <string>

using namespace std;

class Student {

private:

string name;

int group;

public:

void setName(string newName) {

name = newName;

}

string getName() {

return name;

}

};

int main()

{

Student\* S = new Student;

S->setName("Alice");

cout << "Student's name: " << S->getName() << endl;

delete S;

return 0;

}

15) #include <iostream>

#include <string>

using namespace std;

class Student {

private:

string name;

int group;

public:

void setName(string newName) {

name = newName;

}

string getName() {

return name;

}

};

int main()

{

Student S;

S.setName("Alice");

cout << "Student's name: " << S.getName() << endl;

return 0;

}

16) s.name;

17) #include <iostream>

#include <string>

using namespace std;

class Student {

public:

char\* name;

int group;

};

int main()

{

Student\* S=new Student;

S->name;

}

**Вывод:** Задача была выполнена. Всё получилось.