МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

По лабораторной работе № \_\_

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

Проверил\_:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Брест – 202\_\_

**Ход работы**

**ЧАСТЬ 1**

**ТАБЛИЦА UML**

|  |  |  |
| --- | --- | --- |
| << class >>  Circle |  | << class >>  Line |
| -int r |  | -int l |
| << Конструкторы >>  +Circle()  +Circle(int r)  +Circle(const Circle& obj)  << Методы >>  +void TO\_SET(int r)  +void TO\_SHOW()  +int TO\_GET\_VAL()  +int& TO\_GET\_VAR()  +float SQUARE()  +float LENGTH()  +Circle& operator = (const Circle& obj)  << Друзья >>  +friend float SECTOR(Circle& obj, int n)  +friend float SEGMENT(Circle& obj, int n)  +friend void DOUBLE(Circle& obj) |  | << Конструкторы >>  Line()  Line(int l)  Line(const Line& obj)  << Методы >>  void TO\_SET(int l)  void TO\_SHOW()  int TO\_GET\_VAL()  int& TO\_GET\_VAR()  Line& operator = (const Line& obj)  << Друзья >>  friend void EXPAND(Line& obj, int n)  friend void SHORTEN(Line& obj, int n)  friend void STACK\_LINES(Line& obj1, Line& obj2) |
| Класс для хранения информации об одном круге и простых вычислений: площадь круга и длинна окружности. |  | Класс для хранения информации об одной линии. |
|  |  |  |
| << class >>  Square |  |
| -int r |  |
| // Конструкторы  Square()  Square(int r)  Square(const Square& obj)  // Методы  void TO\_SET(int r)  void TO\_SHOW()  int TO\_GET\_VAL()  int& TO\_GET\_VAR()  float SQUARE()  float PERIMETER()  Square& operator = (const Square& obj)  friend void INCREASE(Square& obj, int n)  friend void DECREASE(Square& obj, int n)  friend void STACK\_SQUARES(Square& obj1, Square& obj2) |  |
| Класс для хранения информации об одном квадрате и простых вычислений: площадь квадрата и периметр квадрата. |  |

**РЕАЛИЗАЦИИ МЕТОДОВ**

|  |
| --- |
| **Circle** |
| // Конструкторы  Circle::Circle() { r = 1; }  Circle::Circle(int r) { this->r = r; }  Circle::Circle(const Circle& obj) { this->r = obj.r; }  // Методы  void Circle::TO\_SET(int r) { this->r = r; }  void Circle::TO\_SHOW() { cout << r << endl; }  int Circle::TO\_GET\_VAL() { return r; }  int& Circle::TO\_GET\_VAR() { return r; }  float Circle::SQUARE() { return pi \* r \* r; }  float Circle::LENGTH() { return 2 \* pi \* r; }  Circle& Circle::operator = (const Circle& obj) { this->r = obj.r; return\* this; }  // Friend  float SECTOR(Circle& obj, int n) { return pi \* pow(obj.r, 2) \* n / 180; }  float SEGMENT(Circle& obj, int n) { return 0.5 \* obj.r \* (pi \* n / 180 - sin(180 / pi \* n)); }  void DOUBLE(Circle& obj) { obj.r \*= sqrt(2); } |
| **Line** |
| // Конструкторы  Line::Line() { l = 0; }  Line::Line(int l) { this->l = l; }  Line::Line(const Line& obj) { this->l = obj.l; }  // Методы  void Line::TO\_SET(int l) { this->l = l; }  void Line::TO\_SHOW() { cout << l << endl; }  int Line::TO\_GET\_VAL() { return l; }  int& Line::TO\_GET\_VAR() { return l; }  Line& Line::operator = (const Line& obj) { this->l = obj.l; return \*this; }  // Friend  void EXPAND(Line& obj, int n) { obj.l \*= n); }  void SHORTEN(Line& obj, int n) { obj.l /= n); }  void STACK\_LINES(Line& obj1, Line& obj2) { obj1.l = obj1.l + obj2.l; } |
| **Square** |
| // Конструкторы  Square::Square() { a = 0; }  Square::Square(int l) { this->a = l; }  Square::Square(const Square& obj) { this->a = obj.a; }  // Методы  void Square::TO\_SET(int a) { this->a = a; }  void Square::TO\_SHOW() { cout << a << endl; }  int Square::TO\_GET\_VAL() { return a; }  int& Square::TO\_GET\_VAR() { return a; }  float Square::SQUARE() { return a \* a; }  float Square::PERIMETER() { return a \* 4; }  Square& Square::operator = (const Square& obj) { this->a = obj.a; return \*this; }  // Friend  void INCREASE(Square& obj, int n) { obj.a \*= n); }  void DECREASE(Square& obj, int n) { obj.a /= n); }  void STACK\_SQUARES(Square& obj1, Square& obj2) { obj1.a = sqrt(obj1.SQUARE() + obj2.SQUARE()); } |

**MAIN и ВЫВОД**

|  |  |
| --- | --- |
| int main()  {  Circle R1(10);  R1.TO\_SHOW();  cout << SECTOR(R1, 30) << endl;  cout << SEGMENT(R1, 30) << endl;  DOUBLE(R1);  R1.TO\_SHOW();  cout << SECTOR(R1, 30) << endl;  Line AB(15), BC(145);  AB.TO\_SHOW();  BC.TO\_SHOW();  EXPAND(AB, 4);  SHORTEN(BC, 6);  AB.TO\_SHOW();  BC.TO\_SHOW();  STACK\_LINES(AB, BC);  AB.TO\_SHOW();  Square ABCD(15), PSTU(145);  ABCD.TO\_SHOW();  PSTU.TO\_SHOW();  INCREASE(ABCD, 4);  DECREASE(PSTU, 6);  ABCD.TO\_SHOW();  PSTU.TO\_SHOW();  STACK\_SQUARES(ABCD, PSTU);  ABCD.TO\_SHOW();  } |  |