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**

**ТАБЛИЦА UML MAIN**

|  |  |  |
| --- | --- | --- |
| Weapon |  | #include "Pistol.h"  #include "Rifle.h"  int main()  {  srand(time(NULL));    Pistol USP(10, 0.456);  cout << USP.SHOOT() << endl;  Rifle M4(28, 0.634);  cout << M4.SHOOT() << endl;  USP.ADD\_SCOPE();  M4.ADD\_SCOPE();  M4.ADD\_LENGTH(97);  cout << USP.SHOOT() << endl;  cout << M4.SHOOT() << endl;  } |
| #int damage  #float accuracy |  |
| << Конструкторы >>  +Weapon()  +Weapon(int damage, float accuracy)  +Weapon(const Weapon& obj)  << Методы >>  +void TO\_SET\_DAMAGE(int damage)  +void TO\_SET\_ACCURACY(int accuracy)  +void TO\_SET(int damage, float accuracy)  +void TO\_SHOW()  +int TO\_GET\_VAL\_DAMAGE()  +int& TO\_GET\_VAR\_DAMAGE()  +float TO\_GET\_VAL\_ACCURACY()  +float& TO\_GET\_VAR\_ACCURACY()  +int SHOOT() |  |
| Класс-родитель для реализации общих основных команд для создания классов различного вида вооружения. |  |

**ВЫВОД**

|  |  |  |
| --- | --- | --- |
| Pistol |  |  |
| #bool is\_scope |  |
| << Конструкторры >>  +Pistol()  +Pistol(int damage, float accuracy)  +Pistol(const Pistol& obj)  << Методы >>  +void ADD\_SCOPE()  +void REMOVE\_SCOPE()  +int SHOOT() |  |
| Класс для реализации особенностей модификации и стрельбы из винтовки. |  |
|  |  |
| Rifle |  |  |
| -int length |  |  |
| << Конструкторы >>  +Rifle()  +Rifle(int damage, float accuracy)  +Rifle(const Rifle& obj)  << Методы >>  +void ADD\_LENGTH(int l)  +void REMOVE\_LENGTH(int l)  +int SHOOT() |  |  |
| Класс для реализации особенностей модификации и стрельбы из пистолета. |  |  |

**РЕАЛИЗАЦИИ МЕТОДОВ**

|  |
| --- |
| **Weapon** |
| // Конструкторы  Weapon::Weapon() { damage = 10; accuracy = 0.33; }  Weapon::Weapon(int damage, float accuracy) { this->damage = damage; this->accuracy = accuracy; }  Weapon::Weapon(const Weapon& obj) { this->damage = obj.damage; this->accuracy = obj.accuracy; }  // Методы  void Weapon::TO\_SET\_DAMAGE(int damage) { this->damage = damage; }  void Weapon::TO\_SET\_ACCURACY(int accuracy) { this->accuracy = accuracy; }  void Weapon::TO\_SET(int damage, float accuracy) { this->damage = damage; this->accuracy = accuracy; }  void Weapon::TO\_SHOW() { cout << damage << ' ' << accuracy; }  int Weapon::TO\_GET\_VAL\_DAMAGE() { return damage; }  int& Weapon::TO\_GET\_VAR\_DAMAGE() { return damage; }  float Weapon::TO\_GET\_VAL\_ACCURACY() { return accuracy; }  float& Weapon::TO\_GET\_VAR\_ACCURACY() { return accuracy; }  int Weapon::SHOOT() { if (float(rand() % 100) / 100 >= accuracy) return damage; } |
| **Pistol** |
| // Конструкторы  Pistol::Pistol() : Weapon() { is\_scope = false; }  Pistol::Pistol(int damage, float accuracy) : Weapon(damage, accuracy) { is\_scope = false; }  Pistol::Pistol(const Pistol& obj) : Weapon(obj) { this->is\_scope = obj.is\_scope; }  // Методы  void Pistol::ADD\_SCOPE() { is\_scope = true; }  void Pistol::REMOVE\_SCOPE() { is\_scope = false; }  int Pistol::SHOOT() {  double r = double(rand() % 100) / 100;  r += (is\_scope) ? 0.1 : 0;  cout << "Pistol shot " << r << ' ' << accuracy << endl;  if (r >= accuracy)  return damage;  return 0;  } |
| **Rifle** |
| // Конструкторы  Rifle::Rifle() : Pistol() { length = 0; }  Rifle::Rifle(int damage, float accuracy) : Pistol(damage, accuracy) { length = 0; }  Rifle::Rifle(const Rifle& obj) : Pistol(obj) { length = 0; }  // Методы  void Rifle::ADD\_LENGTH(int l) { length = l; }  void Rifle::REMOVE\_LENGTH(int l) { length = l; }  int Rifle::SHOOT() {  double r = double(rand() % 100) / 100;  r += (is\_scope) ? 0.1 : 0;  r += length \* 0.001;  cout << "Rifle shot " << r << ' ' << accuracy << endl;  if (r >= accuracy)  return damage;  return 0;  } |