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**

|  |
| --- |
| << namespace >>  WEB |
| |  | | --- | | << class >>  Client | | -double Cpkg[10] | | + Client ()  +friend void GET\_PKG\_FOR\_FROM(class Client & get, const class Server& send)  +friend void GET\_PKG\_FOR\_FROM(class Server& get, const class Client & send) | | Класс для имитации передачи пакетов клиентом на сервер. | |  | | << class >>  Server | | -double Spkg[10] | | +friend void GET\_PKG\_FOR\_FROM(class Client & get, const class Server& send)  +friend void GET\_PKG\_FOR\_FROM(class Server& get, const class Client & send) | | Класс для имитации передачи пакетов сервером клиенту. | |

|  |
| --- |
| << namespace >>  NOTHING |
| |  | | --- | | << class >>  Unnecessary | | -int why[1000] | | + Unnecessary()  +void Useless\_method() | | Класс абсолютно ненужный, выполняющий что-то ненужное. | |

|  |
| --- |
| << namespace >>  JUST\_ONE\_ANOTHER\_CLASS |
| |  | | --- | | << class >>  ImClass | | -int number | | +ImClass()  +void EXCEPTION(int x) | | Класс для хранения и деления. | |

**РЕАЛИЗАЦИИ МЕТОДОВ**

|  |
| --- |
| **Client & Server** |
| void GET\_PKG\_FOR\_FROM(class Client& get, const class Server& send) {  for (int i = 0; i < 10; i++)  if (double(rand() / 100) / 100 >= 0.4)  get.Cpkg[i] = send.Spkg[i];  for (int i = 0; i < 10; i++) {  try {  if (get.Cpkg[i] == 0)  throw "CLIANT: Lost package";  }  catch (const char\* messange) {  try {  if (i == 0 || i == 9)  throw "CLIANT: Lost package";  get.Cpkg[i] = trunc(((get.Cpkg[i - 1] + get.Cpkg[i + 1]) / 2) \* 100) / 100;  cout << setw(43) << messange << " #" << i << endl;  }  catch (const char\* messange) {  try {  if (i == 0)  throw "CLIANT: Lost package";  get.Cpkg[i] = trunc((get.Cpkg[i - 1] \* 6 / 5) \* 100) / 100;  cout << setw(43) << messange << " #" << i << endl;  }  catch (const char\* messange) {  get.Cpkg[i] = trunc((get.Cpkg[i + 1] \* 5 / 6) \* 100) / 100;  cout << setw(43) << messange << " #" << i << endl;  }  }  }  }  cout << setw(25) << "CLIANT: ";  for (int i = 0; i < 10; i++)  cout << setw(7) << get.Cpkg[i];  cout << endl;  }  void GET\_PKG\_FOR\_FROM(class Server& get, const class Client& send) {  for (int i = 0; i < 10; i++)  if (double(rand() / 100) / 100 >= 0.4)  get.Spkg[i] = send.Cpkg[i];  else  get.Spkg[i] = 0;  for (int i = 1; i < 9; i++) {  try {  if (get.Spkg[i] == 0)  throw "SERVER: Lost package";  }  catch (const char\* messange) {  try {  if (i == 0 || i == 9)  throw "SERVER: Lost package";  get.Spkg[i] = trunc(((get.Spkg[i - 1] + get.Spkg[i + 1]) / 2) \* 100) / 100;  cout << setw(43) << messange << " #" << i << endl;  }  catch (const char\* messange) {  try {  if (i == 0)  throw "SERVER: Lost package";  get.Spkg[i] = trunc((get.Spkg[i - 1] \* 6 / 5) \* 100) / 100;  cout << setw(43) << messange << " #" << i << endl;  }  catch (const char\* messange) {  get.Spkg[i] = trunc((get.Spkg[i + 1] \* 5 / 6) \* 100) / 100;  cout << setw(43) << messange << " #" << i << endl;  }  }  }  }  сout << setw(25) << "SERVER: ";  for (int i = 0; i < 10; i++)  cout << setw(7) << get.Spkg[i];  cout << endl;  }  } |
| **Unnecessary** |
| Unnecessary::Unnecessary() {  for (int i = 0; i < 100; i++)  why[i] = rand() / 101;  }  void Unnecessary::Useless\_method() {  cout << "Useless\_output: ";  for (int i = 0; i < 5; i++) {  int random = rand() / 201;  try {  if (random >= 100)  throw "error";  cout << why[random] << ' ';  }  catch (...) {  cout << "error" << ' ';  }  }  cout << endl;  } |
| **ImClass** |
| ImClass::ImClass() { number = rand() \* rand(); }  void ImClass::EXCEPTION(int x) {  try {  if (x == 0)  throw "zero";  cout << number << " / " << x << " = " << number / x << endl;  }  catch (...) {  cout << number << " / " << x << " = infinity" << endl;  }  } |

**MAIN и ВЫВОД**

|  |
| --- |
| int main() {  srand(time(NULL));  WEB::Client LOC;  WEB::Server SERV;  for (int i = 0; i < 5; i++) {  GET\_PKG\_FOR\_FROM(SERV, LOC);  GET\_PKG\_FOR\_FROM(LOC, SERV);  }  NOTHING::Unnecessary DontKnow;  DontKnow.Useless\_method();  JUST\_ONE\_ANOTHER\_CLASS::ImClass im\_not\_class;  im\_not\_class.EXCEPTION(10);  im\_not\_class.EXCEPTION(0);  } |
|  |