AbstractClass class Game class Product Atributes public: private: Product():Product; _price: double; ~Product():Product; _name: std::string; toString()const =0:virtual std::string; Methods public: Game():Game; Game(std::string, double):Game; class Physical virtual ~Game();Game; & getName() const:virtual const std::string; **Atributes** Interface setName(const std::string &name):virtual void; private: class IEspecificationsDiscount virtual double getPrice() const:virtual double; _tax: double; Methods setPrice(const double price):virtual void; _game=new Game(): Game*; public: toString() const override:std::string; _itemWeight:double; IEspecificationsDiscount():IEspecificationsDiscount; _productDimensions:std::string; ~IEspecificationsDiscount():IEspecificationsDiscount Methods toString()const=0: virtual std::string; public: Physical():Physical; Physical(double,double,std::string, Game*):Physical; ~Physical():Physical; getGame()const:virtual Game* ; setGame(Game*):virtual void; getProductDimensions()const:virtual std::string; virtual double getItemWeight()const; class ICalculusTax Interface getTax()const:virtual double; class ICalculusDiscount Methods setProductDimensions(std::string)virtual void; public: Methods setItemWeight(double):virtual void; ICalculusTax():ICalculusTax; ICalculusDiscount():ICalculusDiscount; setTax(double):virtual void; ~ICalculusTax():ICalculusTax; ~ICalculusDiscount():ICalculusDiscount; calculatePriceWithTaxGame() const override:virtual double; calculatePriceWithTaxGame() const = 0:virtual double; calculateDiscountGame()const = 0:virtual double; toString()const:virtual std::string; toString()const=0:virtual std::string; toString()const=0:virtual std::string; ′1..n class Streaming **Atributes** class Digital private: private: _tax:double; _tax:double; _game=new Game():Game*; _discount:double; Methods _game=new Game(): Game*; public: Methods Streaming():Streaming; public: Streaming(double, Game*):Streaming; Digital():Digital; ~Streaming():Streaming; Digital(double,double,Game*):Digital; getGame()const:virtual Game*; ~Digital():Digital; setGame(Game*):virtual void; getDiscount()const: virtual double ; getTax() const:virtual double; getTax()const:virtual double ; setTax(double tax):virtual void; calculatePriceWithTaxGame()const override:virtual double; setDiscount(double):virtual void; setTax(double):virtual void; getGame()const:virtual Game*; 1..n setGame(Game*):virtual void; calculatePriceWithTaxGame()const override: virtual double; calculateDiscountGame()const override:virtual double; toString()const:virtual std::string; 1..n 1..n class Recive **ATributes** private: _email:std::string; _digital=new Digital():Digital*; 1..n _streaming= new Streaming():Streaming*; _physical=new Physical(): Physical*; _listDigital:std::vector<Digital>; _listStreaming:std::vector<Streaming>; _listPhysical: std::vector<Physical>; 1..n Methods public: Recive(std::string):Recive; Recive(std::string, Digital*const, Streaming* const, Physical* const):Recive; ~Recive():Recive; getEmail()const:virtual std::string; setEmail(std::string):virtual void; getPhysical()const:virtual Physical*; setPhysical(Physical*):virtual void; getDigital()const:virtual Digital*; setDigital(Digital*):virtual void; getStreaming()const:virtual Streaming*; setStreaming(Streaming*):virtual void; addPhysical(Physical*):virtual void; addDigital(Digital*):virtual void; addStreaming (Streaming*):virtual void; save(const std::string &filename):virtual void; toString()const:virtual std::string:

AbstractClass