**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; \_game=new Game():Game\*; tax:double; discount:double; Methods \_game=new Game(): Game\*; public: Methods Streaming():Streaming; Streaming(double, Game\*):Streaming; public: Digital():Digital; ~Streaming():Streaming; Digital(double,double,Game\*):Digital; getGame()const:virtual Game\*; setGame(Game\*):virtual void; ~Digital():Digital; getDiscount()const: virtual double ; getTax() const:virtual double; getTax()const:virtual double ; setTax(double tax):virtual void; setDiscount(double):virtual void: calculatePriceWithTaxGame()const override:virtual double; 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 1..n class Recive **ATributes** private: Interface email:std::string; 1..n|1..n 1..n class ISave \_digital=new Digital():Digital\*; public: \_streaming= new Streaming():Streaming\*; class Save ISave():ISave; \_physical=new Physical(): Physical\*; public: ~ISave():ISave; \_listDigital:std::vector<Digital>; Save(); addPhysical(Physical\* )=0:virtual void; \_listStreaming:std::vector<Streaming>; ~Save(); addDigital(Digital\*)=0:virtual void; \_listPhysical: std::vector<Physical>; addPhysical(Physical\* )override:virtual void; addStreaming (Streaming\* )=0:virtual void; Methods addDigital(Digital\*)override:virtual void; save(const std::string &filename)=0: virtual void; public: addStreaming (Streaming\* )override:virtual void; Recive(std::string):Recive; //String method Recive(std::string, Digital\*const, Streaming\* const, Physical\* const):Recive; save(const std::string &filename)override: virtual void; ~Recive():Recive; private://Data private getEmail()const:virtual std::string; \_listDigital:std::vector<Digital>; setEmail(std::string):virtual void; \_listStreaming: std::vector<Streaming>; getPhysical()const:virtual Physical\*; \_listPhysical:std::vector<Physical>: std::vector<Physical>; setPhysical(Physical\*):virtual void; \_iPrint=new Print():IPrint\*; getDigital()const:virtual Digital\*; Interface setDigital(Digital\*):virtual void; class IPrint getStreaming()const:virtual Streaming\*; public: setStreaming(Streaming\*):virtual void; IPrint():IPrint; addPhysical(Physical\*):virtual void; ~IPrint():IPrint; | «instanceOf» addDigital(Digital\*):virtual void; addPhysical(Physical\*)=0:virtual void; addStreaming (Streaming\*):virtual void; addDigital(Digital\* )=0:virtual void; |1..\m/1..n 1..n save(const std::string &filename):virtual void; addStreaming (Streaming\*)=0:virtual void; virtual std::string; toString()const:virtual std::string: toString()const=0:virtual std::string; class Print public: Print(); addPhysical(Physical\*)override:virtual void; addDigital(Digital\* )override:virtual void ; addStreaming (Streaming\* )override:virtual void; toString()const override:virtual std::string; private://Data private listDigital:std::vector<Digital>; listStreaming: std::vector<Streaming>; \_listPhysical:std::vector<Physical>: std::vector<Physical>; iPrint=new Print():IPrint\*;

AbstractClass